A. NASER

FAST HEADER MATCHING IN NETWORK PACKETS USING FIELD PROGRAMABLE GATE ARRAYS

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

ANWER SABAH NASER

A MASTER OF SCIENCE THESIS
IN
THE DEPARTMENT OF COMPUTER ENGINEERING

MAY 2021

ATILIM UNIVERSITY 2021

FAST HEADER MATCHING IN NETWORK PACKETS USING FIELD PROGRAMABLE GATE ARRAYS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

BY

ANWER SABAH NASER

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
THE DEPARTMENT OF COMPUTER ENGINEERING

MAY 2021

Approval of the Graduate School of Natural and Applied Sciences, Atılım University.

_____

Prof. Dr. Ender KESKİNKILIÇ
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science Engineering in Electrical and Electronics Atılım University.

_____

Assoc. Prof. Dr. Gökhan ŞENGÜL
Head of Department

This is to certify that we have read the thesis FAST HEADER MATCHING IN NETWORK PACKETS USING FIELD PROGRAMABLE GATE ARRAYS submitted by Anwer Sabah Naser, and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Asst. Prof. Dr. Mehmet Efe ÖZBEK
Supervisor

Examining Committee Members:

Asst. Prof. Dr. Enver ÇAVUŞ
Electrical and Electronics Engineering Department _____
Ankara Yıldırım Beyazıt University

Asst. Prof. Dr. Mehmet Efe ÖZBEK
Electrical and Electronics Engineering Department _____
Atilim University

Asst. Prof. Dr. Hakan TORA
Electrical and Electronics Engineering Department _____
Atilim University

Date: ***7/5/2021***

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Anwer Sabah Naser

Signature: _____

ABSTRACT

FAST HEADER MATCHING IN NETWORK PACKETS USING FIELD PROGRAMABLE GATE ARRAYS

NASER, ANWER SABAH

M.S. of Science, Computer Engineering

Supervisor: Asst. Prof. Dr. Mehmet Efe ÖZBEK

MAY 2021, 48 pages

The hardware architecture of the parallel process multiple RAM that emulates the behaviors of content addressable memory for packet classification is presented in this thesis. With the increase in Internet networks' speed, the speed of detection of intruders has become a basic requirement. In this work, a packet header field is used in a fast and efficient way to detect intruders to prevent them from accessing the data. The application test results were fast and compatible when used the FPGA board technique from Xilinx. Finally, the design, synthesis of this parallel process multiple RAM packet header detector has been achieved using Vivado 2018.2 simulator, and coding is written in Verilog HDL language and Xilinx Artix-7 FPGA (Field Programmable Gate Array) kit was used.

Keywords -: packet classification; pipeline; CAM RAMs, CLBs, HDL

ÖZ

ALAN PROGRAMLANABILIR KAPI DIZILERINI KULLANARAK AĞ PAKETLERINDE HIZLI BAŞLIK EŞLEŞTIRME

NASSER, ANWER SABAH

Y. Lisans . Bilim, Bilgisayar Mühendisliği

Danışman: Dr. Öğr. Üy. Mehmet Efe Özbek

Mayıs 2021, 48 sayfa

Paket sınıflandırması için içerik adreslenebilir belleğin davranışlarını taklit eden paralel işlem çoklu RAM'in donanım mimarisi bu tezde sunulmuştur. İnternet ağlarının hızındaki artışla birlikte, davetsiz misafirlerin tespit edilme hızı temel bir gereklilik haline geldi. Bu çalışmada, verilere erişmelerini önlemek için davetsiz misafirleri tespit etmek için hızlı ve verimli bir şekilde bir paket başlığı alanı kullanılmıştır. Xilinx'in FPGA kart tekniği kullanıldığında uygulama test sonuçları hızlı ve uyumluydu. Son olarak, bu paralel işlem çoklu RAM paket başlık algılayıcısının tasarımı, sentezi Vivado 2018.2 simülatörü kullanılarak gerçekleştirildi ve kodlama Verilog HDL dili ve Xilinx Artix-7 FPGA ile yazılmıştır.

(Field Programmable Gate Array) kiti kullanıldı.

Anahtar Kelimeler -: paket sınıflandırması; boru hattı; CAM RAM'ler, CLB'ler, HDL

DEDICATION

I dedicate my thesis to …….

My beloved parents whose words of encouragement

and push for tenacity ring in my ears.

My brother has never left my side and is

very special.

My many friends who have supported me throughout the

work.

All of you have been my best cheerleaders.

ACKNOWLEDGMENTS

In the name of God, the Most Gracious, the Most Merciful. And prayers and peace be upon our Master Muhammad and his pure family. I would like to express my gratitude and many thanks to my research supervisor, Asst. Prof. Dr. Mehmet Efe ÖZBEK, for his directives, advice, and support that facilitated my work and overcoming the obstacles ahead, I wish him all the best and success, I also thank the staff members of Atilim University for providing facilities. Finally, all thanks, respect, and love to my father Although he is no longer of this world and mother who has done everything to support me from my childhood until this moment and my small family my wife (Arwa Amer) how always supported me with great encouragement and my children Zaid and Rend.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. CHAPTER 1

INTRODUCTION

1.1 Preface

The world today lives in the era of the technological revolution that has entered all aspects of public life and all-important fields, including industrial, medical, and even military. Therefore, special measures were required these days, after the invasion of the Coronavirus and the global trend towards technology and the increasing need to use the transmission of information via computer networks is very important, and the protection of this information along the communication path has become a basic requirement. Recent applications include, for example, college exam questions, financial correspondence, and sensitive economic transactions over the Internet. Internet networks have evolved to reach very high speeds, which requires updating protection means by finding rapid response techniques to detect intruders and prevent them from accessing data sent over the network that could be exploited by thieves and even competitors in some cases. As the value and importance of information increases, the importance of preserving it becomes a basic requirement.

The effort exerted by intruders is also developing. Therefore, information networks faced great and rapid challenges in decoding software codes to obtain data and exploit it to achieve illegal gains. To develop communications security and protect networks, software flaws must be studied and errors corrected to prevent hacking programs and disclose addresses of network users and must be updated These programs are continuously applied to meet the network's requirements for data transfer speed and variety of transmission media in a way that does not negatively affect the quality of the network.

The most types of attacks against computer networks can be briefly classified as:

- protocol-specific attacks: Protocols such as ARP, IP, TCP, UDP, ICMP, and various application protocols can leave openings to inadvertently break into the network. Case in point: Attackers often impersonate protocols or phishing protocol messages to carry out man-in-the-middle attacks and thereby gain access to data that they otherwise cannot access, or to disable target devices on the network.
- Traffic floods: By creating traffic loads that are too large for systems to adequately scan, attackers can create chaos and congestion in network environments, allowing them to carry out attacks without ever being detected
- Malicious Trojan horse program: As the name suggests, Trojans create a network backdoor that gives attackers easy access to systems and any available data. Unlike other viruses and worms, Trojans do not reproduce by infecting other files, and they do not self-replicate. Trojans can be served from online archives and file repositories, and they often originate from peer-to-peer file exchanges.
- worms: One of the easiest and most harmful techniques to infiltrate a network is the common or independent worm virus or virus. Worms often spread via email attachments or instant messages, and they consume large amounts of network resources, preventing authorized activity from taking place. Some worms are designed to steal certain types of confidential information, such as financial information or any personal data related to Social Security numbers, and then transfer that data to attackers who wait outside the organization's network.

Usually, all types of data are sent over the Internet in the form of a series of small packets, each containing a head and a body. The packet header contains the routing and control data, as it contains the IP address of the sender, the IP address of the receiver, a number representing the data exit port at the sender, and a number representing the data receiving port on the receiver, in addition to the transport protocol used in the network.

The body of the package contains the content of the transmitted data and it is a part of the total data to be transmitted. To protect this data during transmission, all incoming packets are tested. Two methods can be used to test the packets, the head test, and the body contents test. In the header test, the source and destination IP address are matching tests with a blacklisted address list to detect any of these addresses to prevent them from accessing the network. In the packet content test, the message content is matching tests with previously stored data from previous attacks to determine the type of attack.

Current work is focused on testing the matching packet header only in protecting network data, by studying previous work techniques in this field and using one of the latest methods, the most efficient in implementing the test matching circuits.

1.2 Problem Statement

Corresponding package address testing can be performed either in software or hardware. Hardware implementation for testing the matching packet header is easier. However, data rate, throughput, and execution speed are increasing in hardware implementation. The software cannot meet the speed requirements of applications in transport and communication networks such as wireless communication systems, radar systems and other applications that need real-time in their operation. On the other hand, various difficulties or challenges appear in the hardware Designs, including the volume of resource use, energy consumption, and hardware circuit size. Therefore, the main challenge of this work is the design and implementation of FPGAs using FPGAs for high-speed execution.

1.3 Literature Survey

Many previous kinds of research related to testing the matching packet header were implemented. This section reviews some of these works:

- In 2012, Tran Ngoc Thinh, and others, presented their work focusing on multi-pattern signature and proposes an FPGA-based deep packet inspection engine for Network Intrusion Detection (NIDS), They stated that their system test result achieved a transfer speed of 1.1 Gbps and worked properly on a real-time network environment [1].

- In 2015, Layla Hattim Abood, The researcher proposed a five-part design for a Ternary Content Addressable Memorie (TCAM) using an FPGA. Each part was examined separately to obtain an RTL scheme for the synthesis fraction of VHDL. The test results were convincing[2].

- In 2018 INAYAT ULLAH, and others, used multi-ported SRAM memory to design and implements a TCAM on FPGA using a multi-pumping to achieve efficient utilization. The depth limit for BRAMs on FPGAs limits the storage efficiency of TCAM bits, so they mapped sections of the TCAM table into sub-blocks of BRAM components and achieved a memory-efficient TCAM. They implemented their design on the Virtex-6 xc6vlx760 FPGA. Moreover, compared to TCAM designs based on current FPGAs, it offers 2.85 times better performance per memory[3].

- In 2018, YEIM-KUAN CHANG and HAN-CHEN CHEN, In this paper, a high-throughput and low-cost connected structure using a new Repetitive Endpoint Cutting (REC) decision tree was proposed. Since the basic buckets are tied to the paper nodes in the decision trees, they consume a large portion of the total memory, also a bucket pressure scheme has been suggested to reduce the double bases. The researchers indicated that with experimental results for the proposed architecture on the Xilinx Virtex-5/6 FPGA, the required mass RAM by REC is much lower than for FPGA-based methods. Likewise, the proposed parallel and striped architecture can accommodate various tables of 20K bases or more, in FPGAs with 1.6 MB of RAM. With dual-port memory, it has the data transfer rate for 40-byte packets, more than 100 Gbps can be achieved[4].

- In 2019, MARCO MONTAGNA, the thesis aimed to validate the potential for a new approach in implementing a growing hierarchy of self-organization map (GH-SOM) for use in classifying packets in the context of Software Defined Networking (SDN). Which features frequent network configuration updates on the go. The proposal was to train a typical neural network and exploit its properties to provide a more efficient solution for classifying packets during operation. The researcher indicated that the results were meeting the set goals as the self-organized hierarchical map is a suitable solution for SDN applications and that the executing system file can deal with large data sets with low-performance losses[5].

- In In 2019 Weiwen Yu, and Others proposed pseudo-TCAM architecture, they used the SRAM-based hardware architecture that simulates the behaviors of TCAM for classifying packets. They used a prefix inclusion coding method to encode the packet header, and encoding rules were set for SRAM-based matches using a bit selection policy. The specified bits from the input key were used to access a base in SRAM for comparison. The results obtained from the header classifier can reach the Throughput of 426 million packets per second[6].

1.4 Thesis Objectives

The aim of this work is to investigate, design, and implement Hardware for testing the matching packet header based on FPGA (Field Programmable-Gate Array) to achieve the best performance. The specific objectives are as follows:

1- Design and implemented the Hardware for testing the matching packet header by using the blocks of RAM with basic logic circuits.

2- Achieve higher throughput, low latency, and high level of instruction and task parallelism

Verilog HDL (Hardware Description Language) is used to design and implement by using "Vivado 2018.2 ", in addition to, Artix-7, project device xc7a100tcsg324-1 development kit is used to implement this design.

1.5 Thesis Outline

The contents of the remaining individual chapters of this project are briefly reviewed:

Chapter Two, " Theoretical Background " gives the following:

- Theory and applications packet header matching.
- Design of hardware packet header matching.
- FPGA design.
- Hardware Description Language.

Chapter Three, " Design and Implementation of packet header matching " this chapter covers the design workflow, methodology, and tools.

Chapter Four, "Results and Discussion" presents the results of the packet header matching designed and performance analysis.

Chapter Five, "Conclusions and Suggestions for Future Work" consists of conclusions and the main ideas for future work.

2 CHAPTER 2

2.1 Theory and applications packet header matching

This FPGA technology is quite powerful because it can be used to create a new BV-TCAM-based packet classification architecture. Ternary Content Addressable Memory (TCAM) and Bit Vector (BV) algorithm are combined with the Bit Vector (BV) Technical Algorithm Architecture (BVTA) to effectively compress the data representations and increase the throughput.

The gate array is recommended due to its ability to change software quickly. In recent days the new FPGA devices such as Xilinx Virtex-7 offer higher speed and a large number of configurable word widths dual ports on chips. In current networks, RAMs are used, but in comparison with TCAM, they are expensive and not scalable. The TCAM performance is also limited by slow speed. As a result, SRAM and FPGA-based TCAMs are used to achieve high speed and high performance in networking application chips.[7]

Packet classification is part of a network intrusion detection system (NIDS). In intrusion detection applications, rules usually consist of 5-tuple headers, like source IP and destination IP. In a full-featured NIDS rule database, signatures can have a variety of different lengths and locations in their packets. However, packet header fields are the same in length and appeared at the fixed locations in the packet. Though, current string-matching algorithms are not easy to extend for the higher level. It is beneficial to separate the header classification process from the string-matching process.

The solution of the two cross-products can be used to determine a complete match. some software-based NIDS scans content first. Besides, the performance of this software is generally poor. Snort is a popular open-source NIDS that employs signatures to detect online malicious activity.

This software-based system cannot keep up with high-speed networks. The system automatically drops traffic when the system overloads, which harms the CPU on

which the application runs. Snort. Packets are scanned in sequence one by one and then the payloads are screened one by one[8].

Some classifiers achieve high accuracy at the cost of high system complexity, high resource usage, or high-power consumption. Some algorithms utilize limited resources but show poor processing resources. Algorithms that are only suitable for software implementations are limited.

We are using the old method of network device classification to sort Ethernet packets in the old way. For networking devices like routers, firewalls, and intrusion detection systems, packet filtering is a very important task. The FPGA-based firewall incorporates this motor.

2.2 Packet Classification Engine (PCE).

PCE feature recursive checking of the multi-dimensional fields of the packet header. This algorithm reduces the system complexity substantially thus leading to a more secure system. PCE works by adjusting the architecture of a central processing unit. PCE Ethernet Packet is examined based on the destination IP address, source IP address, source port, destination port, and protocol fields. You should know the basics of these certain fields before inspecting the content. Four-way test rules are re-coded into 24-bit sub-rules using tree-based algorithm implementation in the architecture. The IC will compare one line of input data per clock cycle[9].

All firewall rules are created and configured by the designer of the firewall. PCE follows all the rules in the theory. It uses "default deny" as its main strategy for preventing all unmarked packets on the network. The proposed PCE uses a complicated piece of algorithm based on decreasing the complexity of the classification rules and changing them to a lower scale.

Table 1 presents the rules used in a packet inspection firewall. Firewall rules are created by combining the field of the packet header. There are 5 rule combinations. 5 factors should be considered.

There will be only one value of output if all the specifications are met, “allow” or “deny”[9]

Table 2-1 Example of firewall rules

Source IP Address	Destination IP Address	Source Port	Destination Port	Protocol	Action
167.205.3.11	167.205.65.32	25	8080	TCP	Allow
192.168.*.*	*.*.*.*	80	*	TCP	Deny
167.205.65.5	*.*.*.*	*	*	UDP	Allow
..*.*	134.25.5.2	>1023	80	TCP	Allow

Classification of network applications (e.g. firewall processing, service quality (QoS) differentiation, private virtual networks, policy routing, and traffic billing, among other value-added services). Next-A FPGA-based parallel architecture for scalable high-speed packages. To provide these services, the router needs to classify packets into different categories based on a set of predefined rules that specify the range of values of the multiple fields in the packet header. A classification of multi-field packets is called this function. Multi-field packet classification has been one of the major challenges for high-speed routers due to the rapid development of the network connection rate and the set size of the rule [10]

The current link rate, for example, has been pushed beyond the OC-768 rate OC stands for optical carrier and the number affixed is the multiple of 51.85 Mbps base rate bandwidth. On a fiber optic carrier, OC-768 supports rates of 40 gigabits per second (Gbps), a rate that translates to the equivalent of the data value of seven CD-ROMs in one second. In the worst case, 40 Gbps, which requires a packet to be processed every 8 ns (where the packets are of minimum size i.e. 40 bytes). Using current software-based solutions to meet the throughput requirements, such throughput is impossible to achieve [11].

recent research in this area seeks to combine algorithmic and architectural approaches, most of which are based on Ternary Content Addressable Memories (TCAMS) Or Various Hashing schemes such as Bloom Filters. However, TCAMs are not scalable with respect to clock rate, power consumption, or circuit area, compared to static random-access memories (SRAMs)[12].

But the main difference between TCAM and SRAM is the main task of a ternary content addressable memory (TCAM) is to search contents stored in its memory. This is done by accessing stored data by content rather than by address; this is unlike static random-access memory (SRAM), which accesses content using the address. TCAM compares the search query with the contents that are preloaded in the entire memory array simultaneously and generates the result. TCAM can be used in one of the three states, as follows: two binary states (0 and 1) and a do not care state (X). Binary states are stored in data cells, and they do not care state is stored in mask cells [13]

2.3 Design of hardware packet header matching

Arief Wicaksana, Arif Sasongko [9] PACKET CLASSIFICATION ENGINE(PCE) Used Build The design is based on requirements. Different types of ethernet packets have 5 or 6 flags. These five things are input to the system - Source IP Address, Destination IP Address, Source Port, Destination Port, Protocol, Start, Reset, and the Clock signal. This design has to be implemented in Figure 2.1.

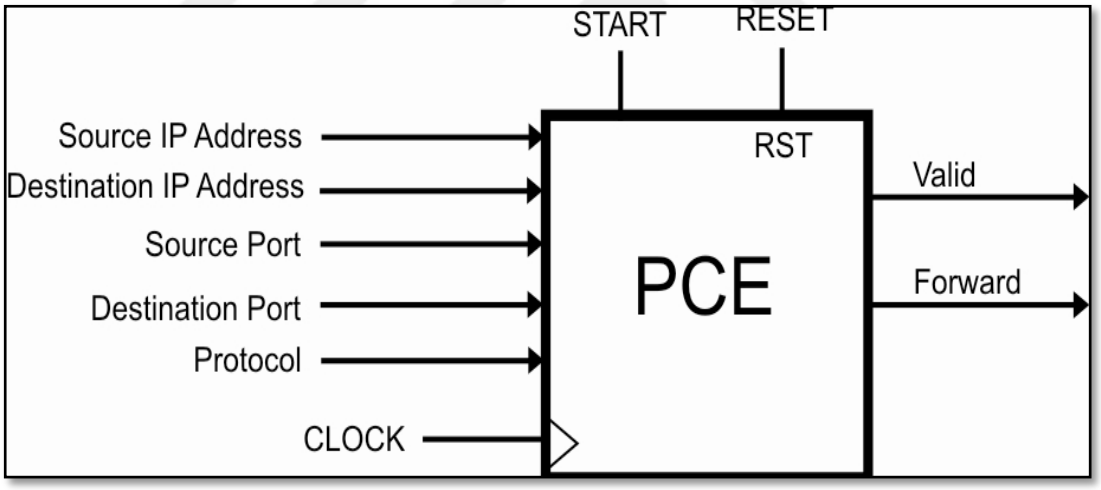

Figure 2.1 Packet Classification Engine (PCE)

Each header must be processed using the process that was designed for the classifi mast look for the best match among a set of rules that are predefined exact or wildcard matching exact is needed for message routing range is typically is used for addressing, while prefix is used for port ranges Since packets may match multiple rules, each rule has a cost. classifi asynchronous hardware-based packet classifiation handles these necessities hardware mechanisms[14]

for many types of applications, the best method is to use associative arrays Ternary content-addressable memories are required for exact, as well as substring matchings. Because embedded peripherals and processing cores becoming more widely available, FPGA devices have been widely adopted for communication and networking applications [15].

there are many studies to deal with (CAMs). Content addressable memories (CAMs) Some RAM (RAM is especially used in applications with a high bandwidth requirement). In contrast to standard RAM: where the user provides data and returns the RAM location, CAM retrieves data (i.e., key). CAM is preferable because it is focused on one goal and less complex than hardware or software-based solutions. The most widely used CAM in network devices is hardware random access memory such as routers and firewalls

CAMs are widely used in networking routers and servers are examples of such devices, the function of which is to classify or classify and drop internet packets) However, CAM has several problems, like high power consumption and low density. manufacturers' kits (FPGA). CAMs come in two forms. A (TCAM) can store 0, 1, and X (don't care a bit). Also, “111XX” will match four values, namely “11100”, “11101”, “11110”, and “11111". When multiple entries have the same content, a priority encoder at the end must select the most important one[16].

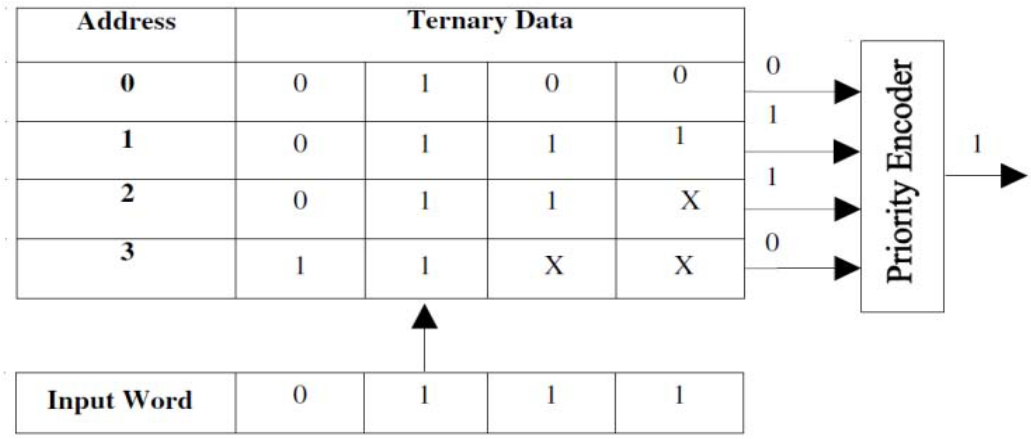

Figure 2.2 Structure of Classical TCAM

The header of each packet should be analyzed for packet classification using various CAMs that are implemented as native CAMs called Application Specific Integrated Circuits (ASIC).

Native CAMs exist several problems.

1- Since a match logic is required, the CAM density is low in each bit and the cost per bit is high.

2- The match logic introduces additional power consumption in each cell and therefore makes CAMs hungry for more power than RAMs.

3- In most Field Programmable Gate Arrays (FPGA), which have broad applications in network devices, native CAMs are not available.[7]

Because of these shortcomings, a more flexible and power-efficient technique is required to emulate CAM behavior. RAM-based CAM emulation has emerged as a promising technique among several solutions, not only because RAM is a relatively mature technology, but also because there are more and greater RAM blocks on modern FPGA. But large RAM also not affected when using large data and addresses

So in this thesis, we will build TCAM based RAM which is found in the FPGA board, and can use special architecture to get the advantages of TCAM and high performance with high speed. When implementing linear search on RAM [17], data words are fetched by the way of using addressing mechanisms. If there are (n) rules stored in RAM, there are at most (n) clock cycles needed to perform a classification by sequentially accessing RAM contents. CAMs search keys are compared bitwise in a single cycle. Implementing such a structure requires high storage and power requirements, which are huge. Finally, TCAMs include XOR for yet another feature, which needs more memory per key bit. Ideally, a priority encoder should be included at the end of the neural network. Existing TCAM chips include the encoding capacity within their program format, so direct read access is difficult to possible. Multipath classification (MMC) feature is needed in modern applications such as intrusion detection and load balancers. Several proposals for updating TCAMs have been made by "re-organizing" maps according to "overlap patterns" or "mapping" neighboring subsets by flipping filters to stand-alone TCAMs. However, under these circumstances, the encoding technique becomes most likely ineffective. This will be evident in the two charts from Figure 2.3 and Figure 2.4.

found. Quick application is the main reason why a CAM is used. The system can also search by reading and comparing all RAM elements one by one. Therefore, RAM search is slower than CAM search.

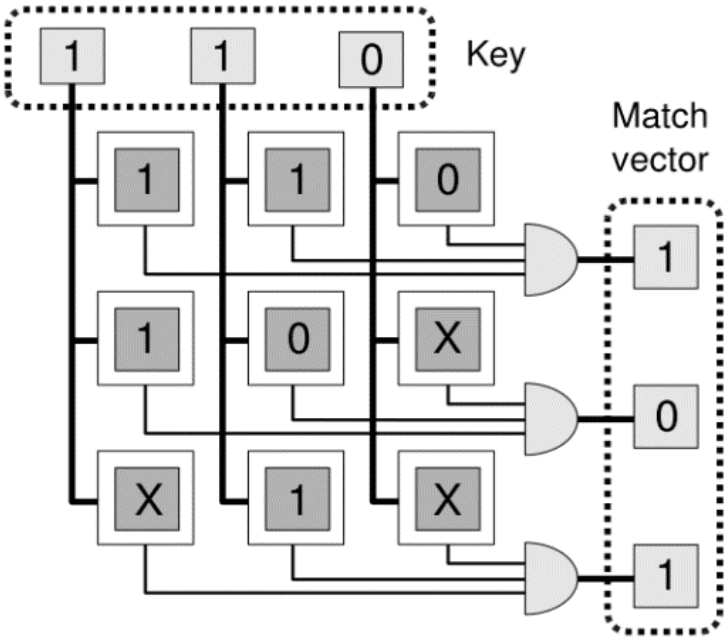

Figure 2.3 Native TCAM access

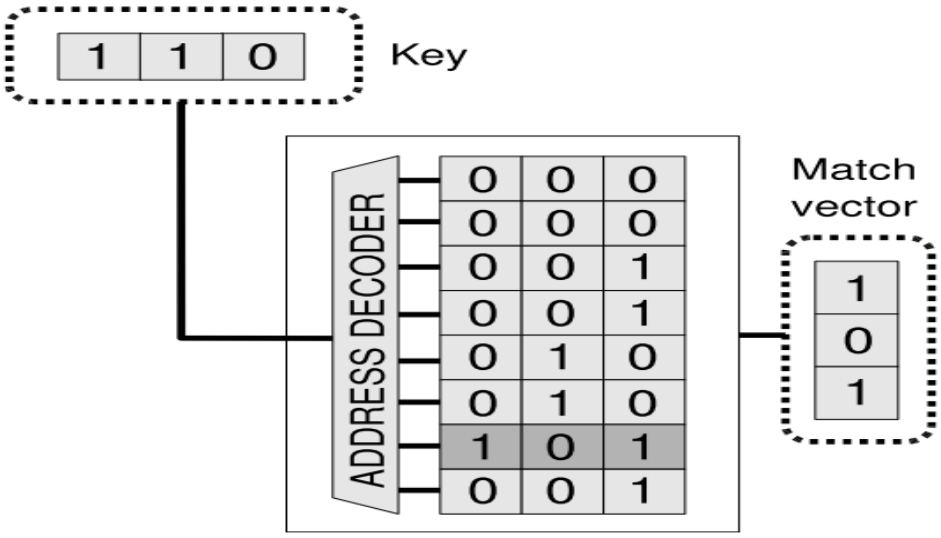

Figure 2.4 emulated TCAM access

Also, RAMs have flaws. CAM requires comparison circuitry to operate, which means we will need more bits to form a comparison operation. The typical TCAM cell consists of two SRAM cells and comparison circuitry.

Recently work Zahid Ullah, Manish K. Jaiswal [18], And In other words, SRAM does a fast search of data which TCAMs does not (SRAMs). The advantages and shortcomings of TCAMs are given. Consequently, the CRAM can behave like TCAM. Ray C. C. Cheung takes SRAM as an abasement to build TCAMs; performs a high-speed search operation compared with static random-access memories (SRAMs). The TCAMs have certain limitations such as slow storage density, relatively slow time of access, low scalability, complex circuits. Therefore, they use the SRAM (with additional logic) configuration to enable it to behave like TCAM and they call it (ZTCAM).

The Xilinx Virtex-7 FPGA has sizes 512 to 36 and 64 to 32. Z-TCAM has been also utilized in 64-to-32 Z-TCAM cell design utilizing the 0.18 μm OSU library to verify the physical and technical feasibility of the Z-TCAM. The entire search took about three clock cycles for each of the designs. The detailed project report was released for both implementations. The ternary Content Addressable Memory allows its memory to be searched based on content, not based on address. Within a set of time, a memory location is sent to the output. A standard transistor–counter cell can store 3 states 0, 1, and X, where X is a non-changing state. Regardless of the data pattern, the corresponding state is always chosen. Because of its similarities to TCP, it is suitable for various applications like router configurations, data compression, anti-virus detection, and image processing[19]

TCAM provides a single clock lookup; however, it has several disadvantages compared with SRAM.

1- TCAM is not subject to intense trade competition in the RAM market.

2- TCAM is not as dense as SRAM.

3- The circuitry of the TCAM cell comparator increases the complexity of the TCAM architecture.

4- Due to massive parallelism, extra logic and capacitive loading lengthen the TCAM access time 3.3 times longer than the SRAM access time [20]

2.4 PREVIOUS WORKS AND MOTIVATIONS

The features and specifications of PACEX are derived from needs. Different types of ethernet packets have 5 or 6 flags. These five things are input to the system - Source IP Address, Destination IP Address, Source Port, Destination Port, Protocol, Start, Reset, and a Clock signal. This design has to be implemented[13]. The TCAM cell consists of two 1-bit SRAM cells (D0 and D1) which store three logic states of the cell. Each SRAM cell consists of cross-coupled inverters and transistors used to transfer information between bits. The pair of transistors becomes a low resistance path from the match line (ML). etched in Figure 2.5. [21] [16]

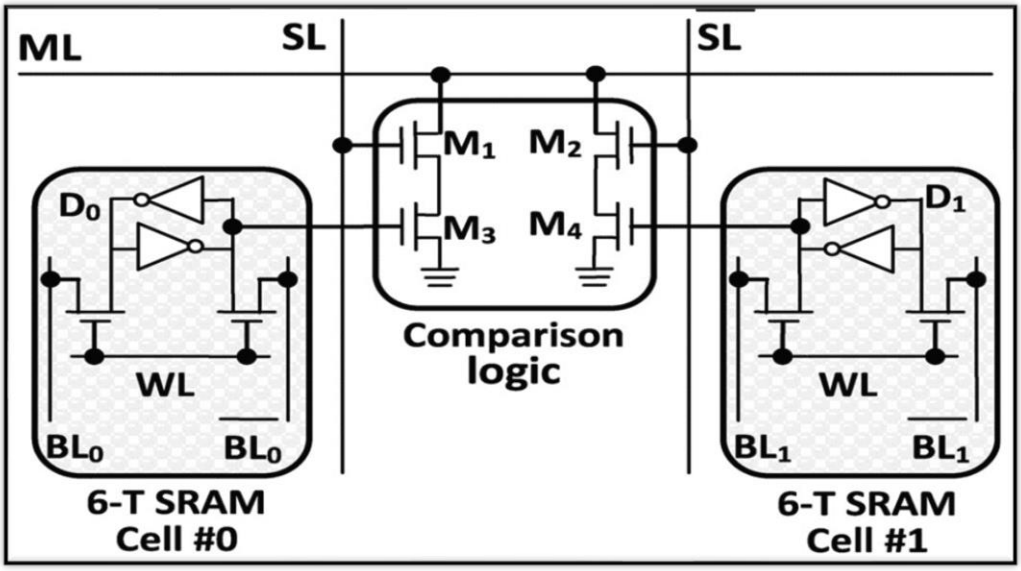

Figure 2.5 The pair of transistors becomes a low resistance path from the match line

On the FPGA platform, various SRAM-based TCAMs have been implemented. A fully programmable TCAM is implemented on the Xilinx FPGA To emulate classic TCAM, it uses Xilinx primitive block RAMs (BRAMs) and distributed RAMs. During a search operation, all of the SRAMs are activated, increasing the overall power consumption. To reduce power consumption, a hierarchical low-power search operation is proposed, in which RAMs are activated hierarchically based on match conditions found in previous RAM blocks.[3], [22][16]. At the moment SRAM-based TCAM has been implemented on the FPGA platform. A scalable TCAM is implemented on the Xilinx Chip to manage huge and complex data. The SRAMs will always stay on while searching. In order to reduce power consumption from[23] a kind of hierarchical search operation based on prior knowledge of RAM addresses has been proposed. SRAM-based TCAM, HP-TCAM [24] Z-TCAM, and E-TCAM [18] All used SRAMs and logic circuits to construct TCAM functionality of packet classifier's software and an FPGA TCAM running on Altera and Xilinx BRAMs, respectively. Another large component of the TCAM is logically subdivided into several relatively small components SRAM-based and logic-based TCAMs were implemented by researchers to construct sophisticated TCAMs. In this way, they partitioned the set of TCAM bits into sets of various sizes and mapped them to SRAM-based TCAM architecture. "The architecture uses two SRAMs of high speed". In the initial period

of system, RAM stores the presence of the QS in the TCAM. SRAM will store the address information (AI) associated with QS HP-TCAM in [25] further stores the index.

2.5 Field-Programmable Gate Arrays (FPGAs)

FPGAs have come a long way in the past five years. Since then, there has been rapid growth in the popularity of consumer electronics like computer laptops, tablet computing, and mobile phones [26][27]

FPGAs have improved due to the development process. When they are reprogrammed, the logic and routing capabilities of their circuits will significantly improve. FPGA (Field-Programmable Gate Array) has become regarded as being among the cheapest new technologies for compilers and verifiers. Many FPGA chips produced by Xilinx don't have any instruction logic written in them. With specific devices, they were able to construct precisely. We will be able to develop a new product because of the discovery of a variation for our old product. This seems to be the strategy of port expansion to follow. [28]

Configuring the FPGA is referred to as "programming the FPGA" FPGAs are becoming incredibly common because hardware and software designers can work together on a single processor board. The project aims for a more general-purpose technology than a microprocessor. Installations must be permanent. The expenses are $10 million to $500 million kinds of and will take several years to develop. The value of improving an ASIC over time can increase. That is why an increasingly larger portion of bigger constructs are controlled by FPGAs. Innovation has had a huge impact on DPI products. [29]

2.5.1 FPGA TECHNOLOGY IN GENERAL

“FPGAs are programmable digital integrated circuits which consist of configurable logic blocks (CLBs) and input/output blocks (IOBs) located around the periphery of the chip”. FPGAs can be used to implement any complex digital circuit as long as you provide the resources. There have been many technological advancements in logic density, chip cost, and overall performance specifications of FPGAs. FPGAs can function at frequencies of up to 500 MHz [30]. Finger 2.6 shown the Overview of FPGA architecture

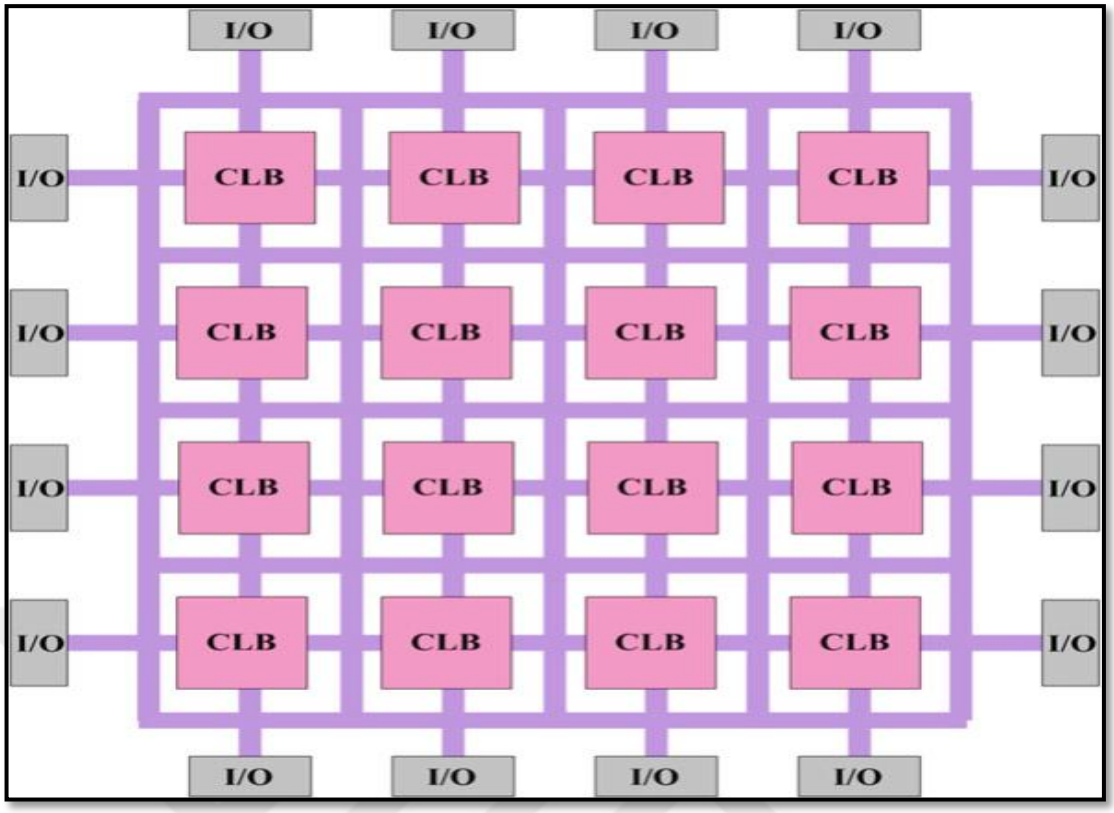

Figure 2.6 Overview of FPGA architecture

2.5.2 Configurable Logic Blocks (CLBs)

Configurable Logical Blocks (CLBs) are an intrinsic component of an FPGA whose core logic provides a storage function for the design of a target application. Configurable logic blocks can be built to perform complex synthesis functions, or simple logic gates can be set up and built such as configurable AND, OR, and XOR logic blocks directly into a community of dimensions that may be linked via programmable routing resources. I/O blocks are set up at the fringes of the community and are more closely related to programmable routing interconnects. The term "programmable/combinable" in FPGAs refers to their ability to implement an entirely new property within a chip after it has been fully manufactured. The recombination/programming of the FPGA is based on the underlying programming technology, which may be geared towards transforming the behavior of the prefabricated partition once it is fabricated [31]

2.5.3 Programming Technologies

Programming technology the programmable routing in FPGA provides connections for the design of a circuit between internal logical blocks and I/O blocks. It is made up of programmable switches and wires which can be configured to obtain the desired connections [31]

There is three programming technology:

- SRAM-based programming technology: FPGAs using SRAM (fixed random access memory) based fixed memory cells. In these devices, most commercial vendors use this technology to program the static memory (SRAM). For configuration, these fixed memory cells are split via an FPGA. In nature, SRAM cells are volatile and must be permanently stored by external devices.
- Technology of Flash Programming: - The use of programming technology based on flash EEPROM (electrically erasable programmable read-only memory) is an alternative to SRAM-based programming technology. The programming technology based on Flash has certain benefits. To avoid an infinite number of reconfigurations, this programming technology is volatile. in flash technology, non-standard CMOS process is used.
- Anti-fuse Programming Technology: - this technology is an substitute for flash-based technologies and SRAM; its low area is the primary one of this technology. Also, in nature, it is non-volatile. "It does not use standard CMOS process" Unable to reconfigure anti-fuse-based devices [32]

2.5.4 Input / Output Blocks

The logic is useful for your computing projects. The FPGA must interface at many different speeds and voltages to link with many different components. Various FPGA devices are available to manufacturers including Altera and Xilinx. Each manufacturer offered different kinds of development boards [33] This research targeted Xilinx Virtex – 7 FPGA because it is affordable at a low cost. And availability in the college laboratories.

2.5.5 TYPES OF FPGA'S

These forms of FPGAs cover the implementation of the logic molecules and the mechanism used to connect the device. The FPGAs are the most commonly used types of reprogrammable SRAM (Multi-time Programmed MTP) and OTP

(One Time Programmed). SRAM-based is the dominant type of FPGA and can be reprogrammed as often as you like. Indeed, when it gets powered up, an SRAM FPGA is reprogrammed, because the FPGA is a fancy memory chip. That's why you want an SRAM FPGA serial PROM or system memory [34]

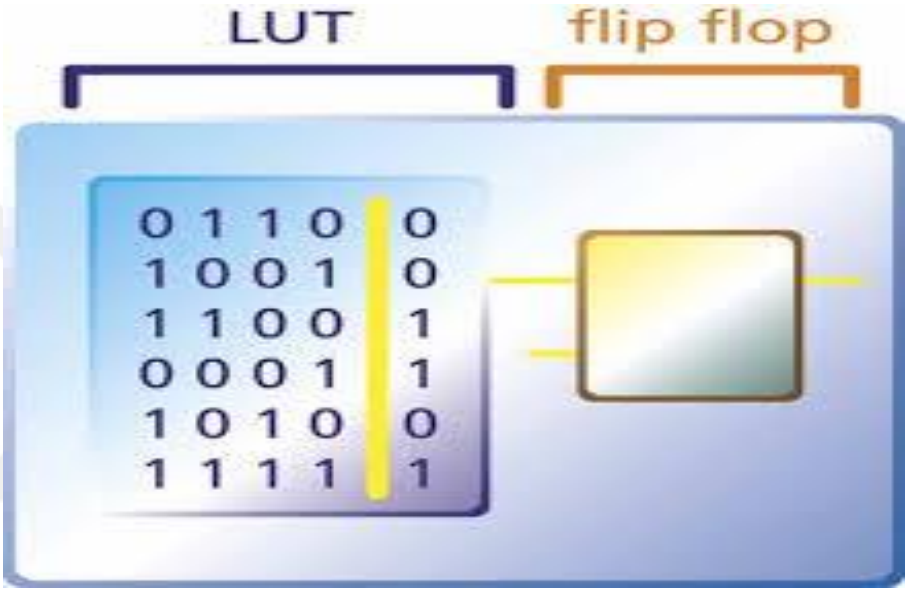

Figure 2.7 SRAM-based reprogrammable.

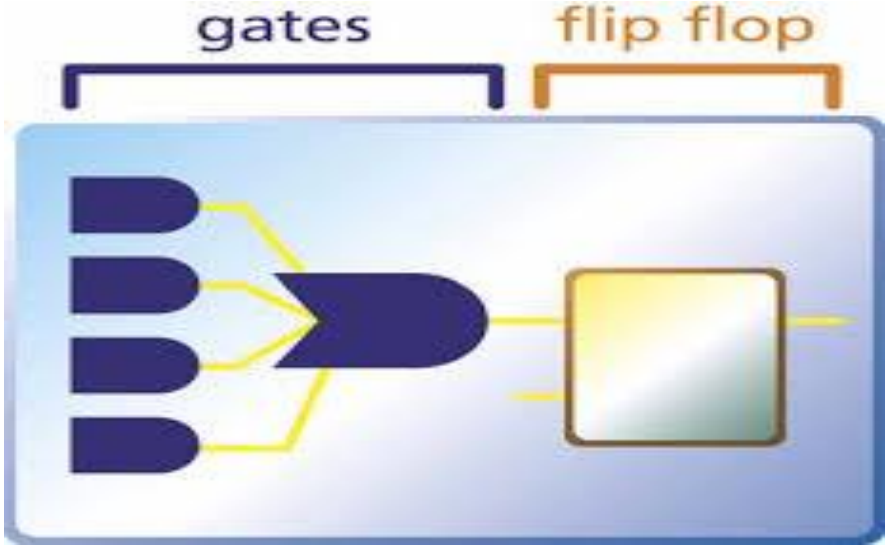

Figure 2.8 One Time Programmed OTP

2.5.6 Xilinx Artix-7 FPGA DDR4

Various companies created various development kits that could be used for prototyping, design, download, and testing. Artix-7 Nexys-4ddr is being used in the current project. Xilinx, a pioneer company in the field of Programming Logic Devices, has presented the aforementioned kit (PLDs). The Artix-7 kit consists of two components: software and hardware. In Xilinx-Vivado 18.2, the software development tool was used. The Nexys-4ddr board is the hardware component, and it is dependent on the Artix-7 FPGA chip. [35]on this board. Diligent Nexys 4 DDR FPGA Trainer boards are shown in Figure 2.9, 2.10.

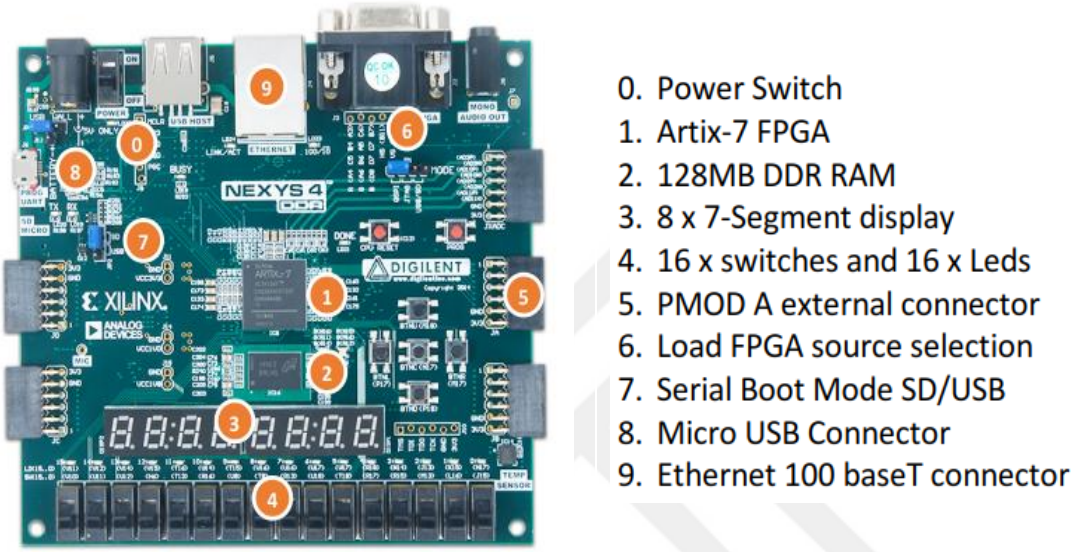

Figure 2.9 NEXYS FPGA Trainer boards components and key elements

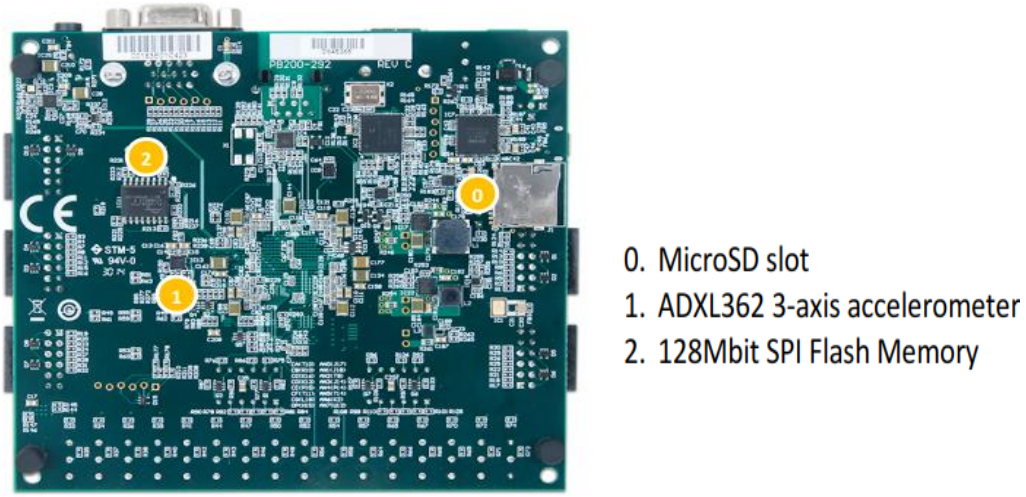

Figure 2.10 Diligent FPGA kit Trainer [36]and key elements

Nexys 4 is a platform that is capable of developing digital integrated circuits and supports many uses for the design and development of digital circuits. Thanks to the FPGA (Xilinx XC7A100T), it can build designs ranging from simple compatible circuits to ultra-fast electronic systems, plus capacity-rich external memory, multiple combinations of USB, Ethernet and other ports, providing a wide variety of features for embedded peripherals. Practicality that does not work perfectly with FPGAs such as the accelerometer, temperature sensor, digital microphone, amplifier, and a large number of input/output devices, use the Nexys 4 for a wide variety of designs.

Features FPGA kit Trainer Xilinx XC7A100T.

- a faster block RAM of 4,866 kilobytes Total block RAM is 4.866 kilobytes.
- Six clock management tiles, each with 240 DSP slices and a phase-locked loop (PLL).16 user switches
- 16M byte Cellular RAM
- Temperature sensor
- Two 4-digit 7-segment displays

The Xilinx Vivado design suite [37] integrates the board and provides constraint files for all hardware links. The trainer's board includes a variety of sensors and connectors, including microphones, sound jacks, VGA25 ports, USB host ports, RGB leads, thermometers, PWM26, and more. The FPGA configuration data is stored with a bit

file extension in files called bitstreams. Xilinx ISE or Vivado users can create and build VHDL, Verilog, or Schematic source files (the EDK is also used for MicroBlazeTM's processor-based embedded designs through the ISE toolkit).

The bitstream is stored in FPGA memory cells based on SRAM. This data defines the logical and circuit connection features of the FPGA and is valid but when the reset button is pressed the board is deleted or a new configuration file is created using the JTAG port.

The Artix-7 100T bitstream is usually 30,607,303 bits and its transfer may take some time. The time required to program the Nexys4 is reduced if the bitstream is compressed prior to programming and the FPGA can then decouple the bitstream itself during setup. Compression ratios of 10x can be achieved depending on the complexity of the design. Within Xilinx Tools (ISE or Vivado), Bitstream compression can be enabled during build. After successful programming, the FPGA will light up the DONE indicator. When the "PROG" button is pressed, the FPGA configuration memory is reset. After the reset, the FPGA immediately tries to reprogram itself from whatever method the jumpers have chosen in programming mode. Figure 2.11 shows the different options available for FPGA configuration. On-board "mode" jumper (JP1) and media select jumper (JP2) select between programming modes.

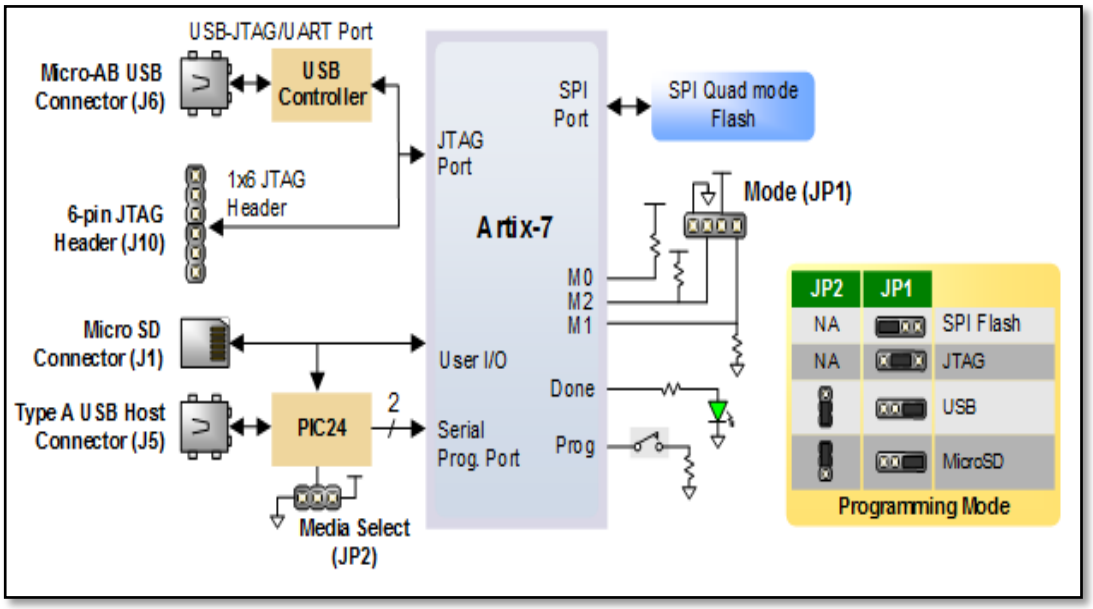

Figure 2.11 The different options available for configuring the FPGA

2.6 Hardware Description Language

FPGA architectures had been intensely investigated during the last decades. A major factor of FPGA architecture studies is the improvement of Computer-Aided Design (CAD) tools for mapping applications to FPGAs. It is nicely installed that the quality of an FPGA-based implementation is essentially decided by the effectiveness of the accompanying suite of CAD tools. Benefits of an in any other case well designed, characteristic wealthy FPGA structure is probably impaired if the CAD tools can not take advantage of the features that the FPGA provides Thus, CAD algorithm studies are essential to the important architectural advancement to narrow the overall performance gaps between FPGAs and different computational devices like ASICs the software flow (CAD flow) takes an application design description in a Hardware Description Language (HDL) and converts it to a stream of bits that is eventually programmed on the FPGA. The process of converting a circuit description into a format that can be loaded into an FPGA can be roughly divided into five distinct steps, namely: synthesis, technology mapping, mapping, placement, and routing. The final output of FPGA CAD tools is a bitstream that configures the state of the memory bits in an FPGA. The state of these bits determines the logical function that the FPGA implements.

Figure 2.12 shows a generalized software flow for programming an application circuit on an FPGA architecture.[38]

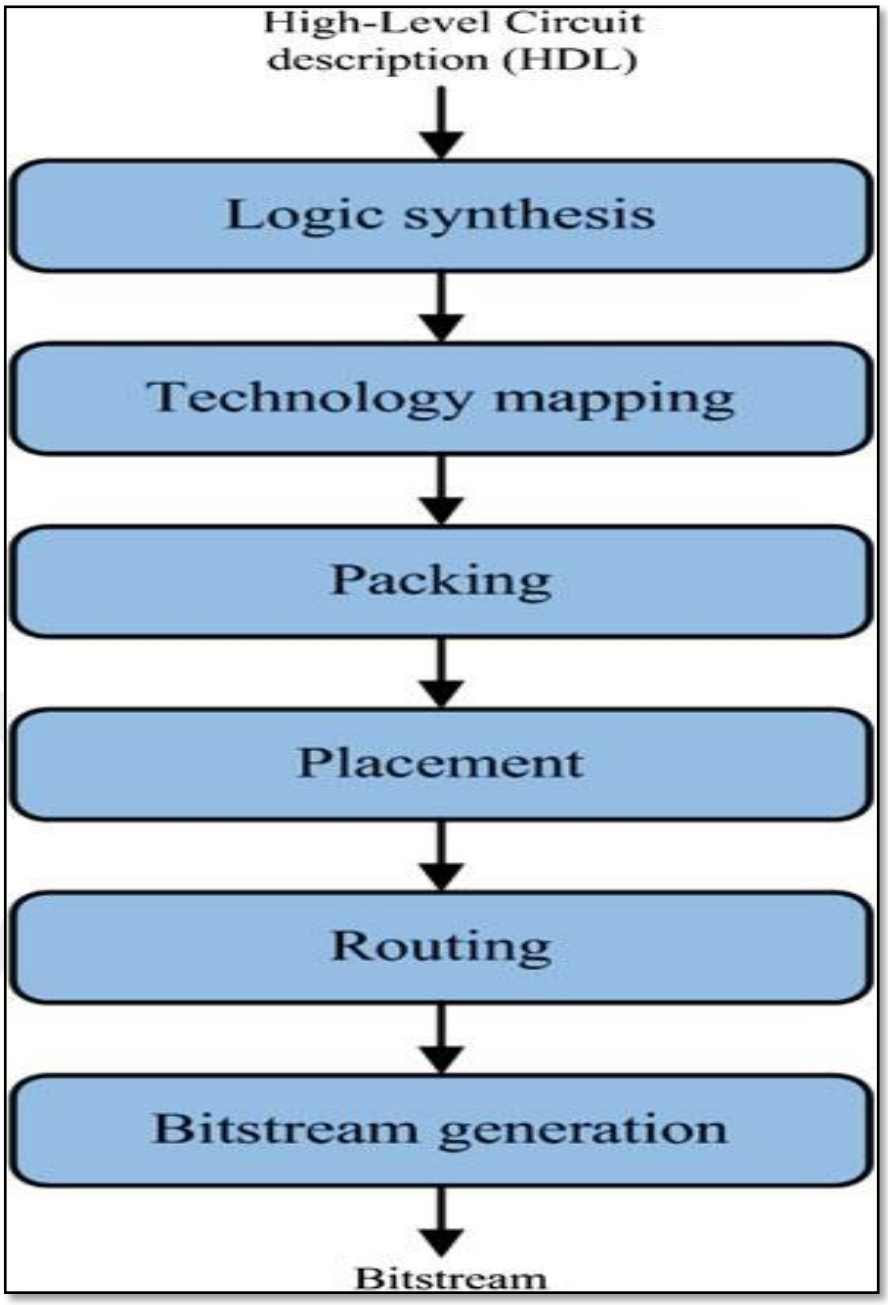

Figure 2.12 FPGA software flow

A Hardware Description Language (HDL) based on design flow involves describing the design (circuit) to be realized in an HDL like the Verilog HDL or VHDL (Very high-speed integrated circuit HDL). HDL is similar to software application programming languages but is used for describing hardware. The functional simulation process simulates the functioning of the circuit capture using one of the above mention design techniques, using functional simulation tools. The input test is applied to the

techniques, using functional simulation tools. The input test is applied to the design during simulation, and the output is verified for the required purpose. The logic synthesis involving optimization and mapping transforms the design into a gate-level netlist corresponding to the logic resource available in the target chip. The chip design may be for the development of a custom ASIC
(Application Specific Integrated Circuit) or for the implementation of required functionalities of the IC in a standard programmable integrated circuit, the FPGA and CPLD (Complex Programmable Logic Device) are an example of the IC in a standard programmable integrated circuit. In the case of CPLD, the logic gates are realized in terms of the gates available in the "macrocells" of the CPLD, whereas for FPGAs the logic gates are realized in terms of "Look Up Tables (LUTs)". The optimization and mapping process optimizes the logic expressions for speed, area, and power [26] "Verilog, standardized as IEEE 1364, is a hardware description language (HDL)" used to version digital systems. It is maximum generally used withinside the layout and verification of digital circuits on the register-switch level of abstraction. It is likewise used in the verification of analog circuits and combined signal circuits, in addition to in "the design of genetic circuits. In 2009, the Verilog standard (IEEE 1364-2005)" become merged into the System Verilog general, developing IEEE Standard 1800-2009. Since then, Verilog is formally a part of the System Verilog language. The modern version is IEEE standard 1800-2017 [39]

Verilog is a hardware descriptive language, meaning it can document and manage the operation of electronic and digital circuits. One of its most prominent uses is Verilog Verify via simulation, for things related to timing testing and analysis. Logical synthesis is another very important application. Verilog helps designers develop and work with different degrees of abstraction. More broadly, Verilog is a language for describing the most frequently used hardware devices, and more than 60,000 active programmers use this language. It can describe a digital system that may include a microprocessor or a slipper. Verilog HDL was chosen for use in this because it can be learned in a shorter time.

3 CHAPTER 3

3.1 METHODOLOGY

This chapter presents the detailed design of a new Hardware architecture to Perform Testing of the Matching Packet Header Based on FPGA. FPGA has been used to implement packet header matching, and according to the use of this new type of FPGA, a new Xilinx tool has been also used, I mean Vivado 2018.2. Xilinx Vivado, is a High-Level Synthesis Kit (HLS), a suite for capturing C ++ based design, simulation, and synthesis to HDL format. The advantage of using this toolkit also reduces effort in HDL design capture and debugging while allowing flexibility in the final implementation of hardware to meet design limitations. Figure 3.1 shown the nexys4 kit that used in this thesis.

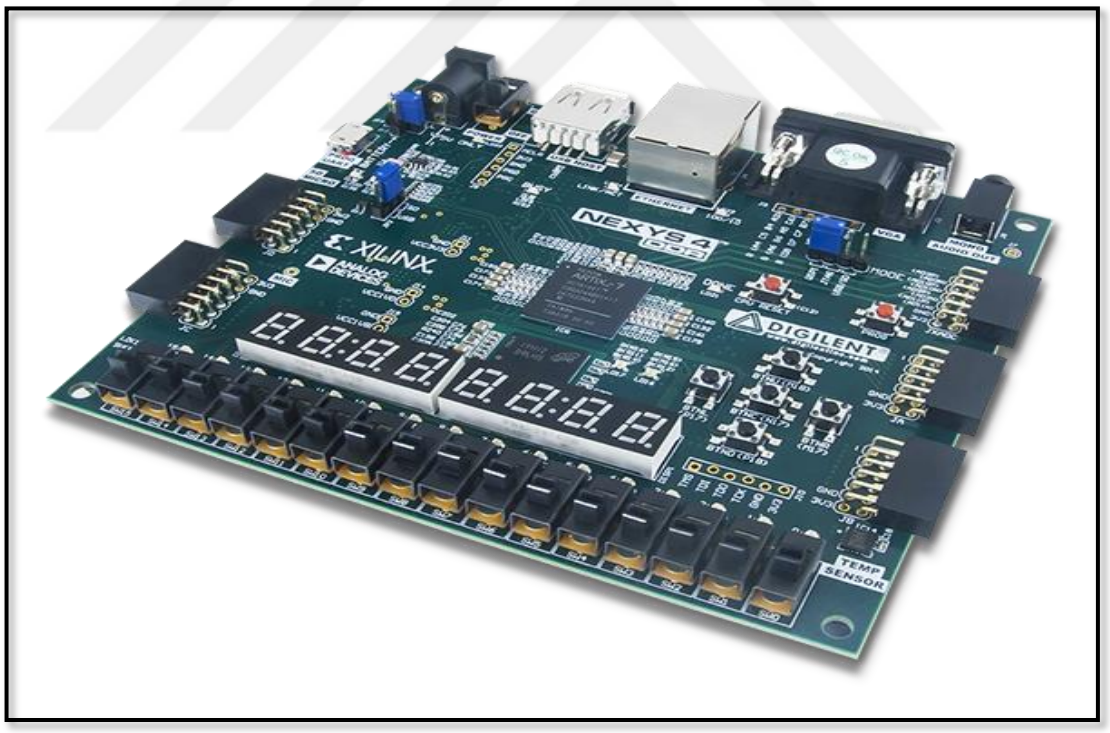

Figure 3.1 Artix-7 Nexys-4ddr kit

3.2 The Development FPGA Kit

Different companies developed various advancement kits for prototyping, developing, importing, and testing purposes. Artix-7 Nexys-4ddr is used in the present work. Xilinx, which is a pioneer company in the field of programming logic devices, introduced the above package (PLDs). There are two important parts of the Artix-7 kit: software and hardware. In Xilinx-Vivado 2018.2, the Software Development Tool was included. The Nexys-4ddr board is the hardware portion and this board is based on the Artix-7 FPGA chip.

After connecting the board to the computer and turning it on, the Artix-7 FPGA must be created or configured (or programmed) before it can be able to perform any function.

1. Through the PC the Digilent USB-JTAG circuit (port J6, labeled "PROG") can be used to program the FPGA any time the power is on.

2. We can transfer files previously stored in Serial Non-volatile Flash (SPI) device to FPGA using SPI port.

3. We can transfer pre-programmed files to FPGA from micro-SD card.

4. By transferring the programming files from a USB memory stick directly that is connected to the USB HID port. The proposed IP matching design based on FPGA RAMs

To take advantage of BRAMs on FPGAs and avoid the circuit complexity and flaws that accompanied others' designs. In this paper, we proposed an IP matching design based on parts of IPv4. They are four parts represented in decimal notation, and each fraction represents a decimal number ranging from 0 to 255 independently of the others. The packet header consists of five fixed fields: source IP address using 4 bytes, destination IP address using 4 bytes, source port number using 2 bytes, destination port number using 2 bytes, and protocols field using 1 byte. Since each byte is independent, it has allocated separate RAM to store its contents, also the 16-bits port number has been divided into two bytes to reduce the size of RAM as shown in Figure 3.2 The flowchart methodology of this thesis is available in Figure 3.3

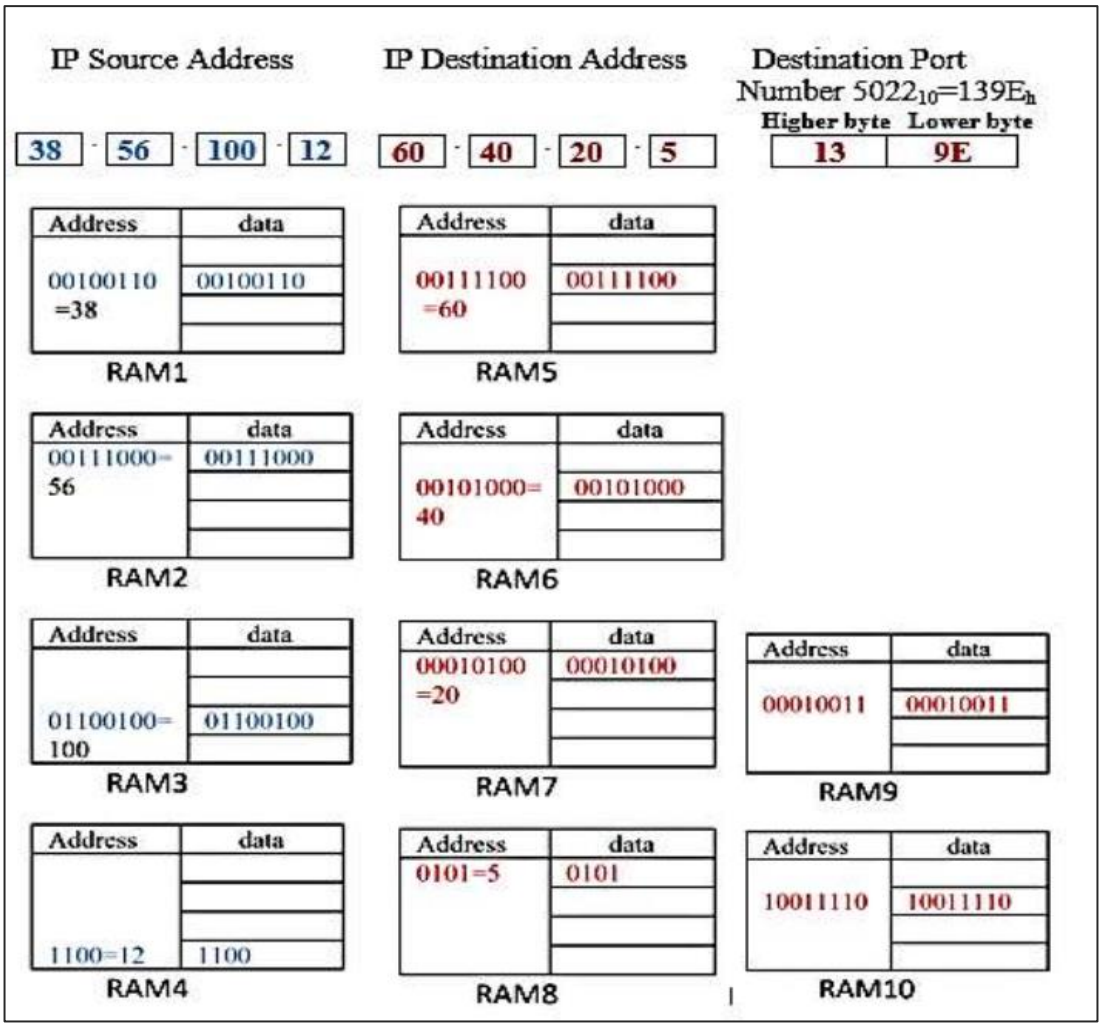

Figure 3.2 10 Distributed RAMs to store all range of IP address and destination numbers.

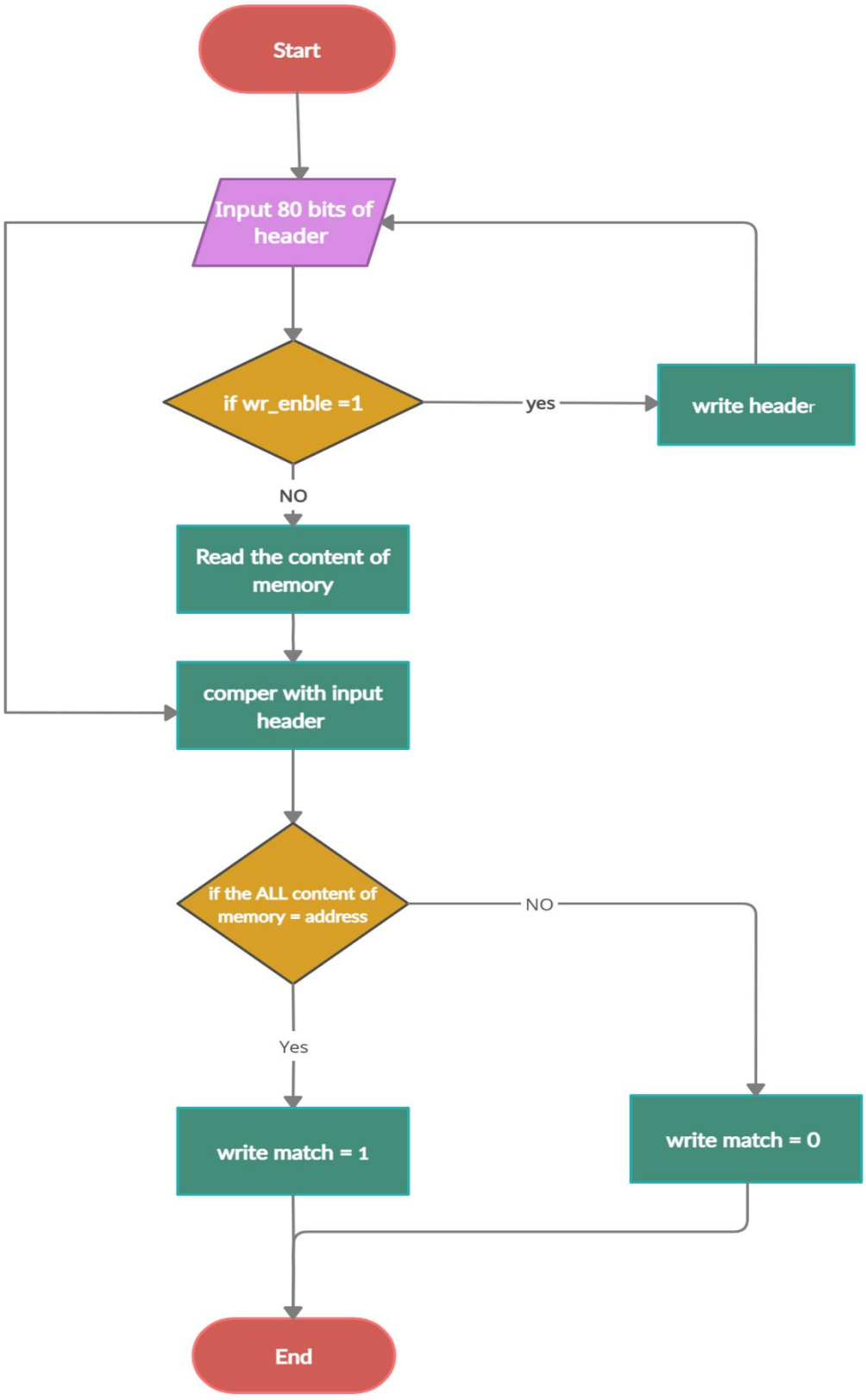

Figure 3.3 The flowchart methodology of program

The specific prefixes addresses have been converted to a number of direct addresses and previously stored into related addressable content RAMs (CAM RAMs) shown in Figure 3.4.

The contents of the packet header have been used as the address to read the content of the distributed CAM RAMs, also used as the input to the comparators that compare these inputs with the RAMs outputs, all the RAMs, and comparators circuits run in parallel. The output of the comparators circuits arrives as input to the AND gate, the input header does not match unless all the comparators' results are matches.

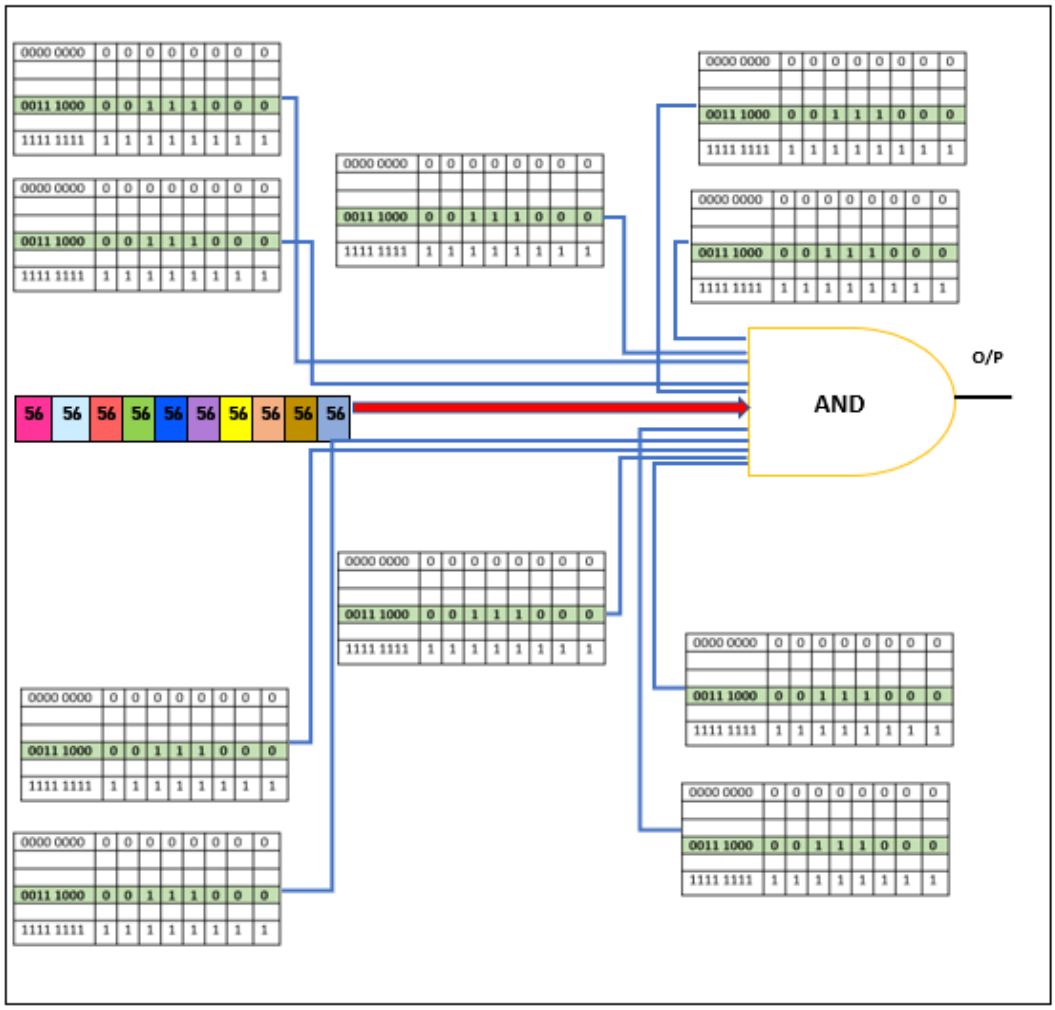

Figure 3.5 10 RAMs compere with I/P header

3.3 Implementation of the proposed packet Header (PH) Matching

In this work, the Vrtix-7[36] FPGA has been used to implement packet header matching, and according to the use of this new type of FPGA, a new Xilinx tool has been also used, I mean Vivado 2018.2. Xilinx Vivado, is a High-Level Synthesis Kit

(HLS), a suite for capturing C++-based design, simulation, and synthesis to HDL format. The advantage of using this toolkit also reduces effort in HDL design capture and debugging while allowing flexibility in the final implementation of hardware to meet design limitations. Figure 3.5, 3.6, 3.7 shows the FPGA implementation of the proposed packet header matching on NEXYS 4DDR board

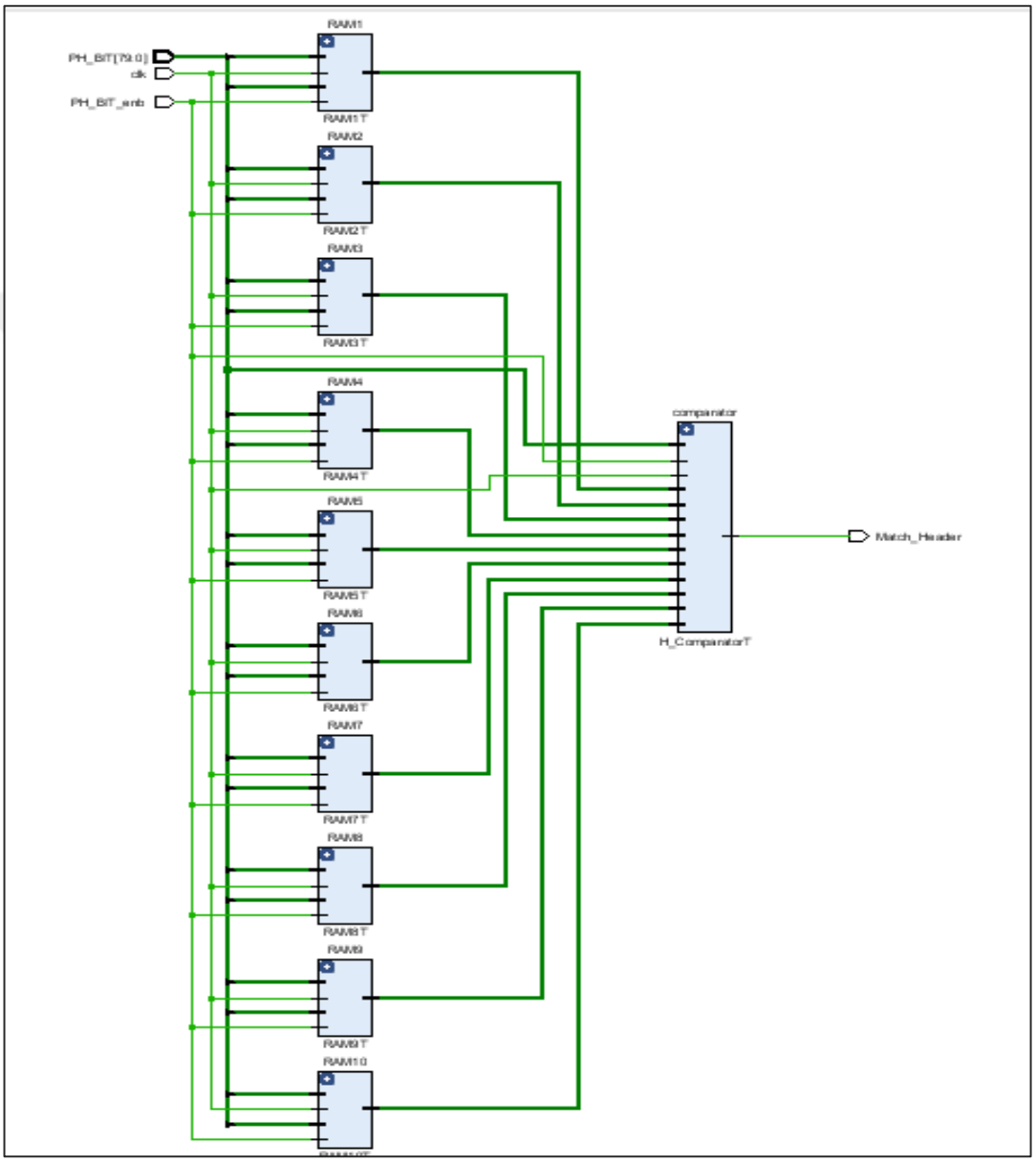

Figure 3.6 FPGA implementation of the proposed packet header matching

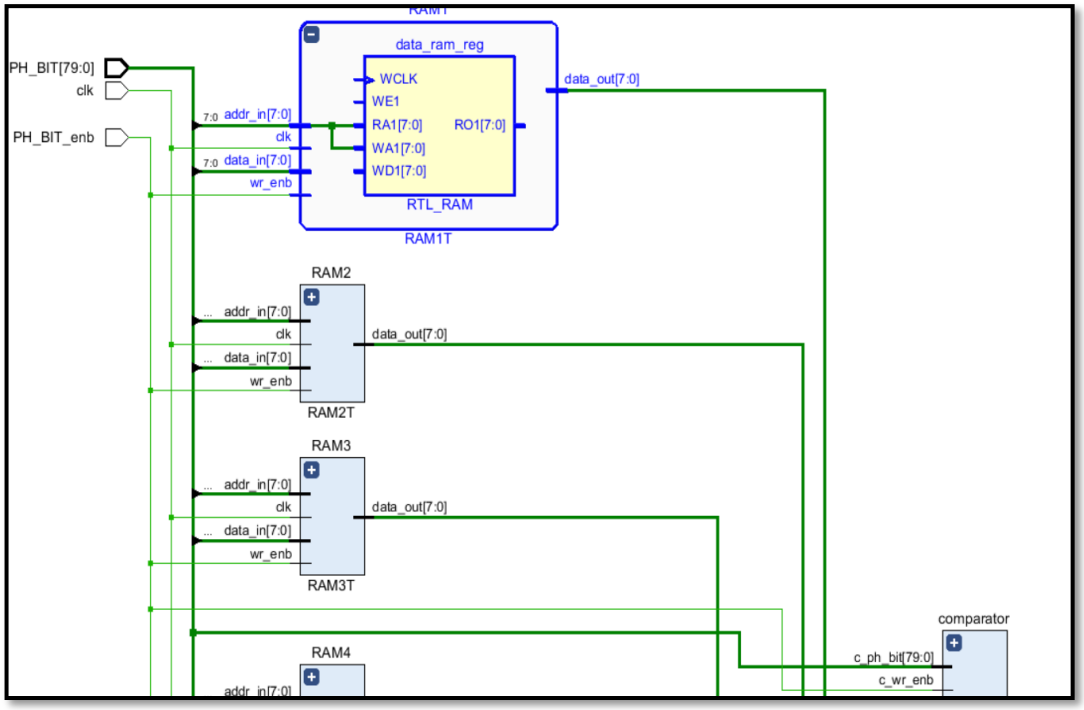

Figure 3.7 Details of CAM RAMs input and output lines

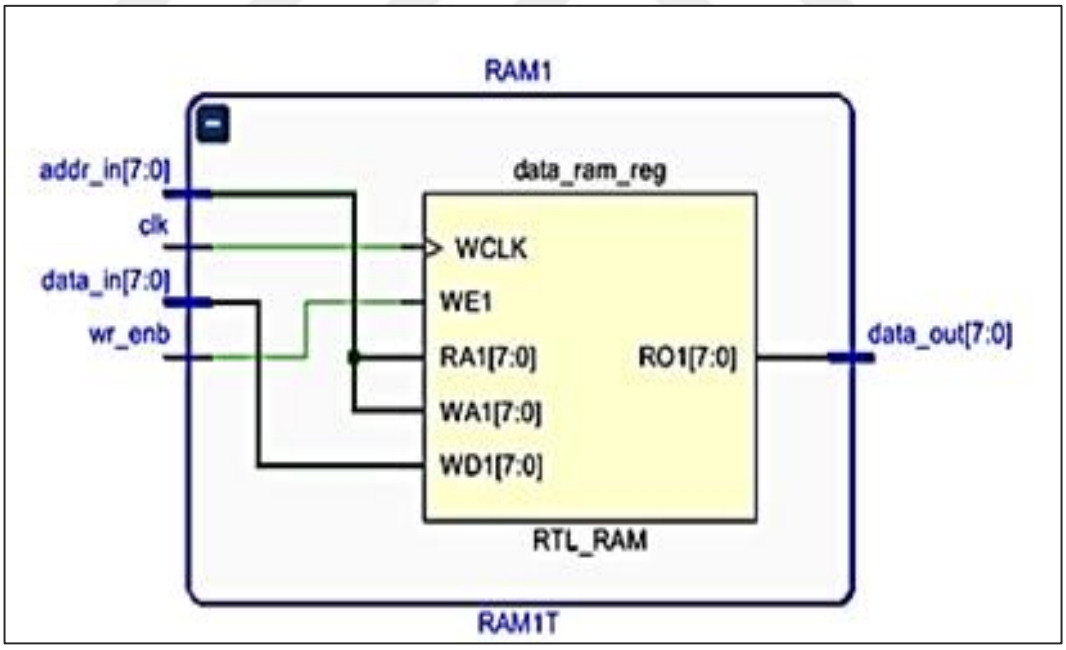

Figure 3.8 Details of the interconnection of RAMs input and output lines

Many research that emulates/implement TCAMs by using RAM but none of them gives correct proof. this new design emulates the size of TCAM by using the same size of RAM depth.

Depth of RAM = depth of TCAM and denoted as (N)

N= number of the word in the TCAM or RAM

The width of TCAM or RAM means the number of bits and denoted as (w)

W= number of bits in the TCAM or RAM

So, the size of the team = N*W.

In this thesis, we have 10 bytes which equal to 80 bits which equals the width of the RAM. so, to take the advantage of using FPGA in parallel we divided RAM into 10 small RAMs each one has a size of 256x8. The address we need it to reach for every location of memory denoted as(D) this detail is shown in Figure 3.8

Where N= 2^D 256 = 2^D , D = 8 for this equation we store each byte in one memory.

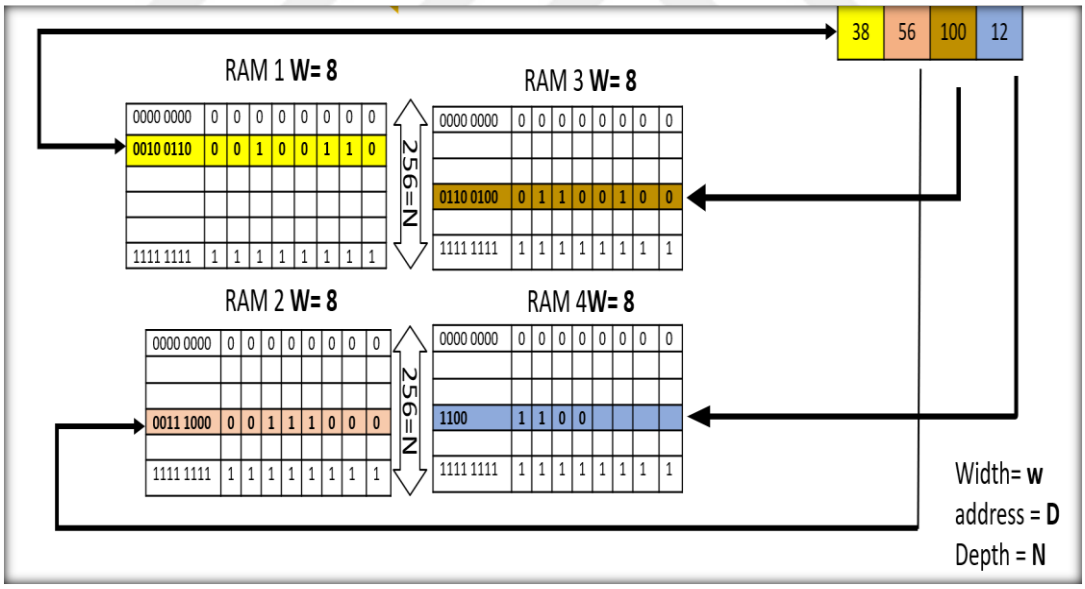

Figure 3.9 Each byte of header store in one memory location have belonged to him

4 CHAPTER 4

IMPLEMENTATION AND SIMULATION RESULTS

4.1 Introduction

The implementation of the proposed FPGA-based packet header matching was presented in this chapter, along with simulation results. The proposed TCAM architecture design for packet header matching was implemented using Xilinx-Vivado and FPGA, with Xilinx-Vivado 18.2 as the simulation environment, as described in Chapter 3. The simulation program was run on a Windows 10 with a 64-bit operating system, Intel Core i7-9th generation 2.60 GHz processor, and 16GB of RAM. The FPGA specifications were Xc7a100tcsg324-1 as a device model, 360 K block RAM, and 100 MHz clock frequency.

4.2 The results

To see if the proposed solution is effective, the system was simulated to verify its functionality and to test the Matching Packet Header Based on FPGA. The proposed design tested with incoming header packet, In the running simulation, 80 bits of input data have been used to define malicious clients that should be blocked. these data stored into the 10 CAM RAMs of size 28x8 bits, the “PH_BIT_enb” bit line has been set to write enable to write into the RAMs memory as the initial setting, then the write enables bit, disabled to start the reading the content of memory for the matching process with the new input data. Figure 4.1,4.2 shows the waveforms for, write-enabled, the input data, and the red color waveform of matching signal that change from mismatch to match when new data identical to that previously stored has been entered. each fraction represents a decimal number ranging from 0 to 255 independently of the others. That means we need to check all contents of each RAM from 0000 0000 to 1111 1111 after converting them from dismal to binary.

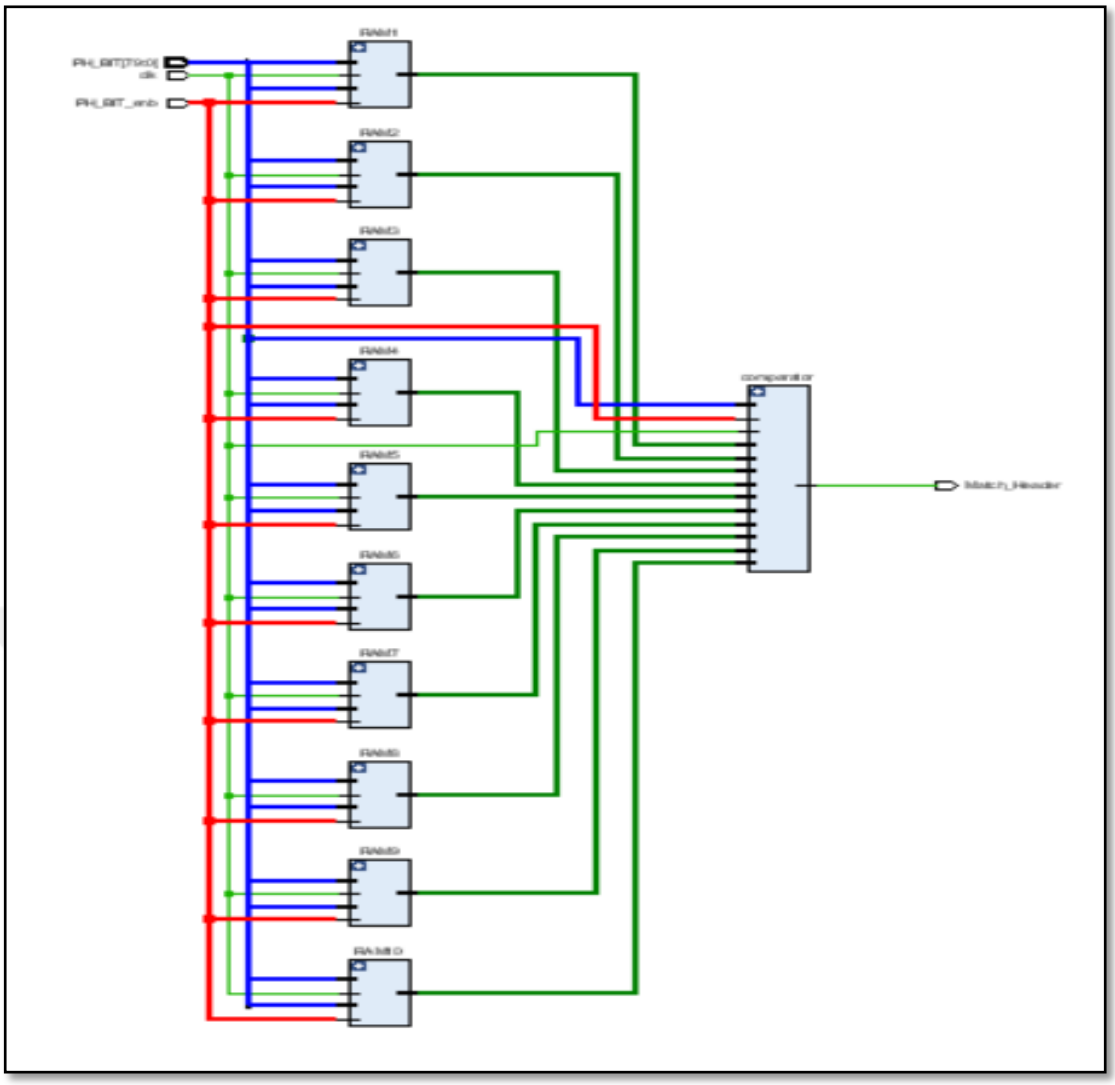

Figure 4.1 The Red link represents the PH_BIT_enb for all RAMs and comparator

The Blue link represents the input header for each RAM and comparator

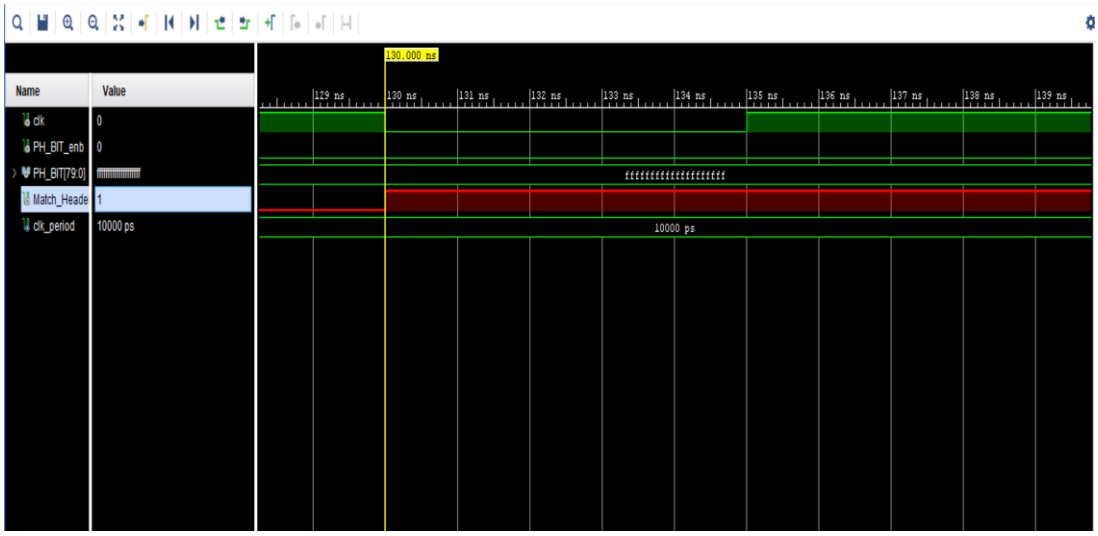

Figure 4.2 Simulation result of (PH_BIT) for match result

Figure 4.3,4.4,4.5 shows the simulation of check the RAM content and address input and the output that belongs to the address with a status of the write enables bit Which is (0) which means the operation is read the content of memory. And the content of all memory locations from 0 to 255.

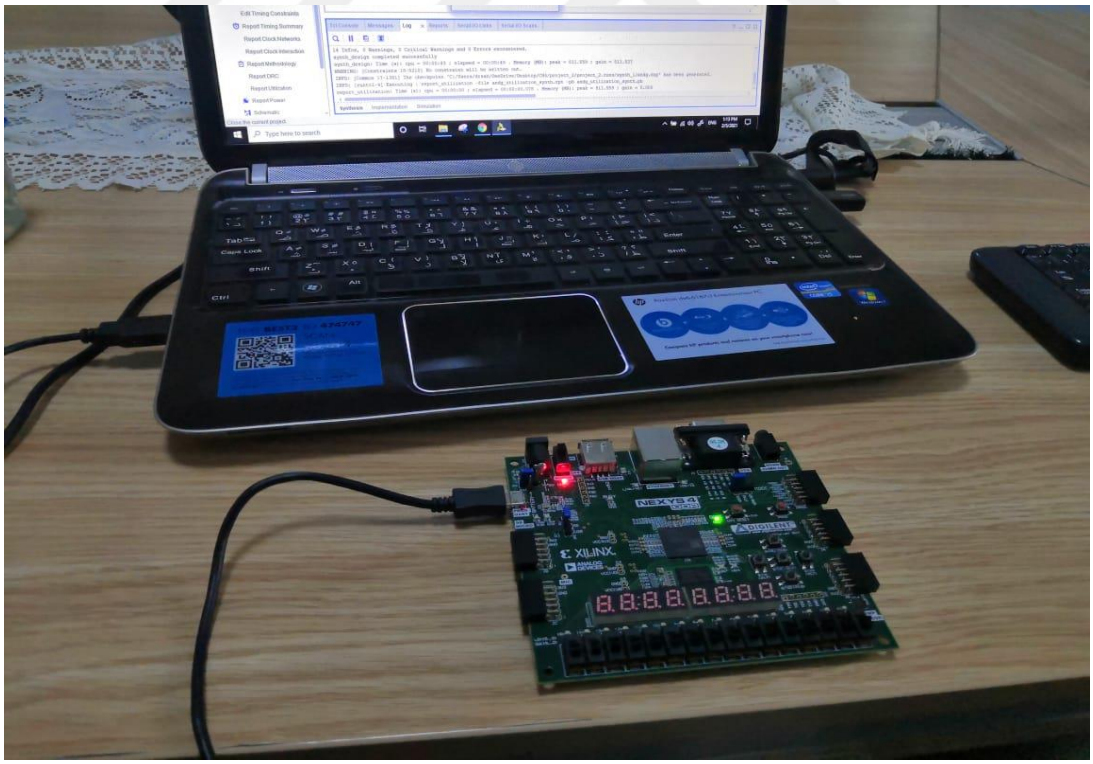

Figure 4.3 FPGA Implementation of proposed packet header testing

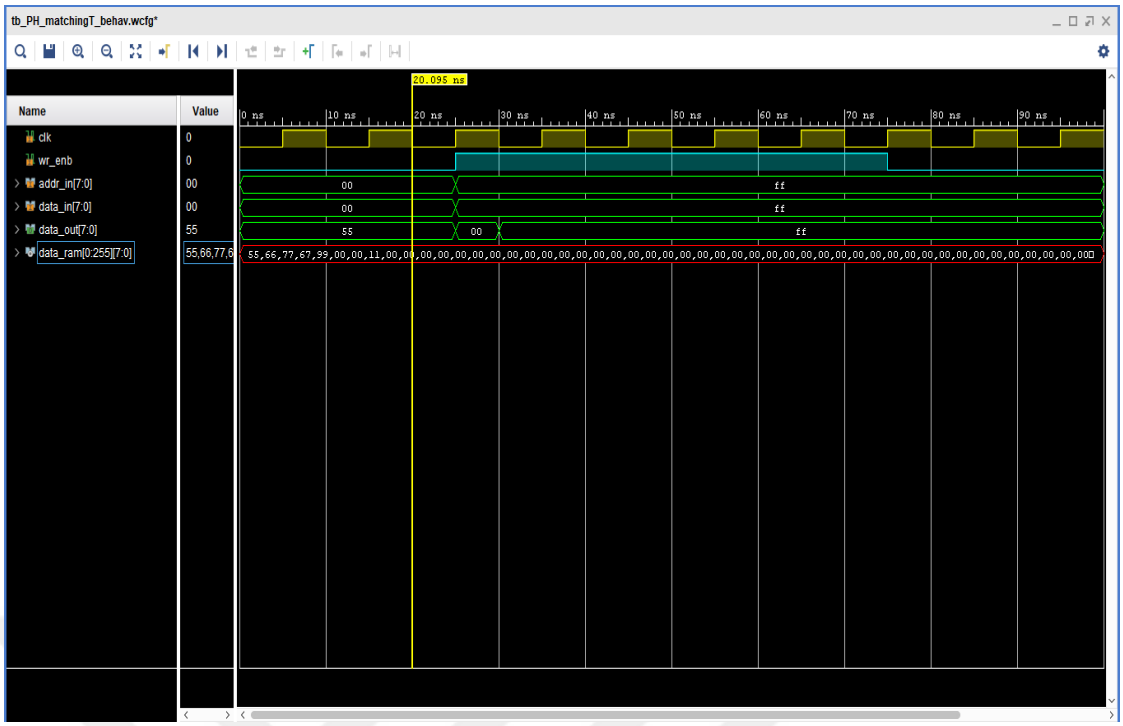

Figure 4.4 Check the RAM content

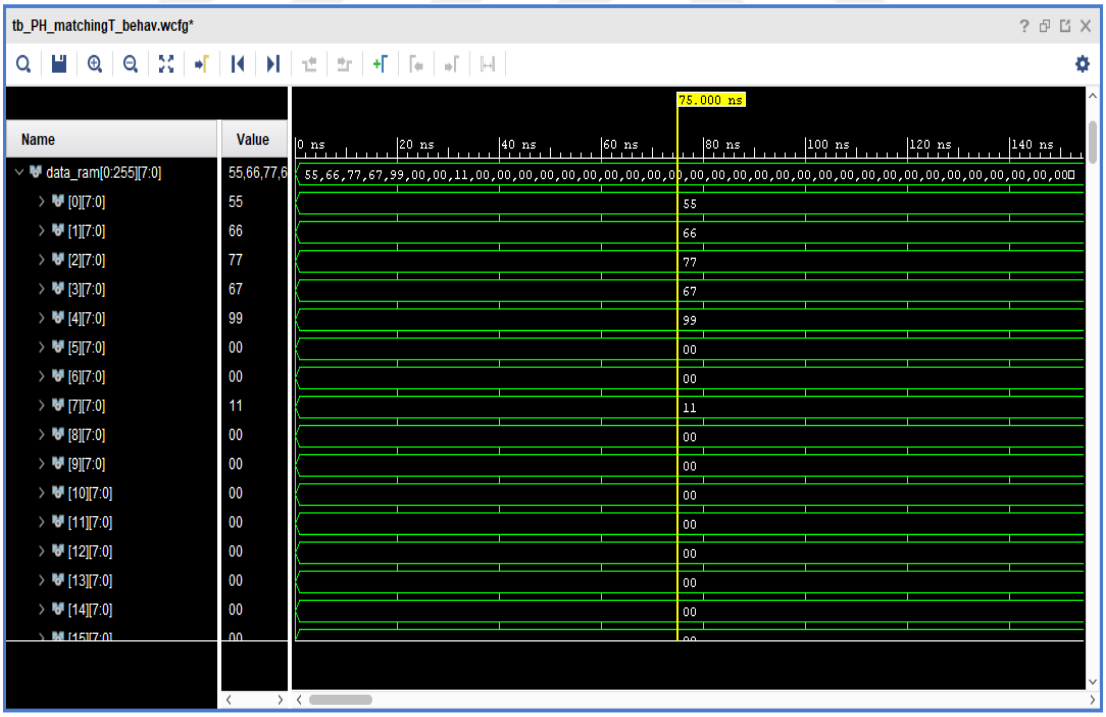

Figure 4.5 The content of all memory locations from 0 to 255

Figure 4.6 shows the simulation result of the different incoming packets which is compere with all content of all RAMs that have the same input address it gives header match when all content of RAMs gives the same result of the header inputs, PH_BIT is 80 bits for the most important parts of the header (source IP, destination IP, port number): source IP address using 4 bytes, destination IP address using 4 bytes, source port number using 2 bytes and each byte equal to 8 bits. So, the 80 bits most equal to all 80 bits stored in the RAMs.

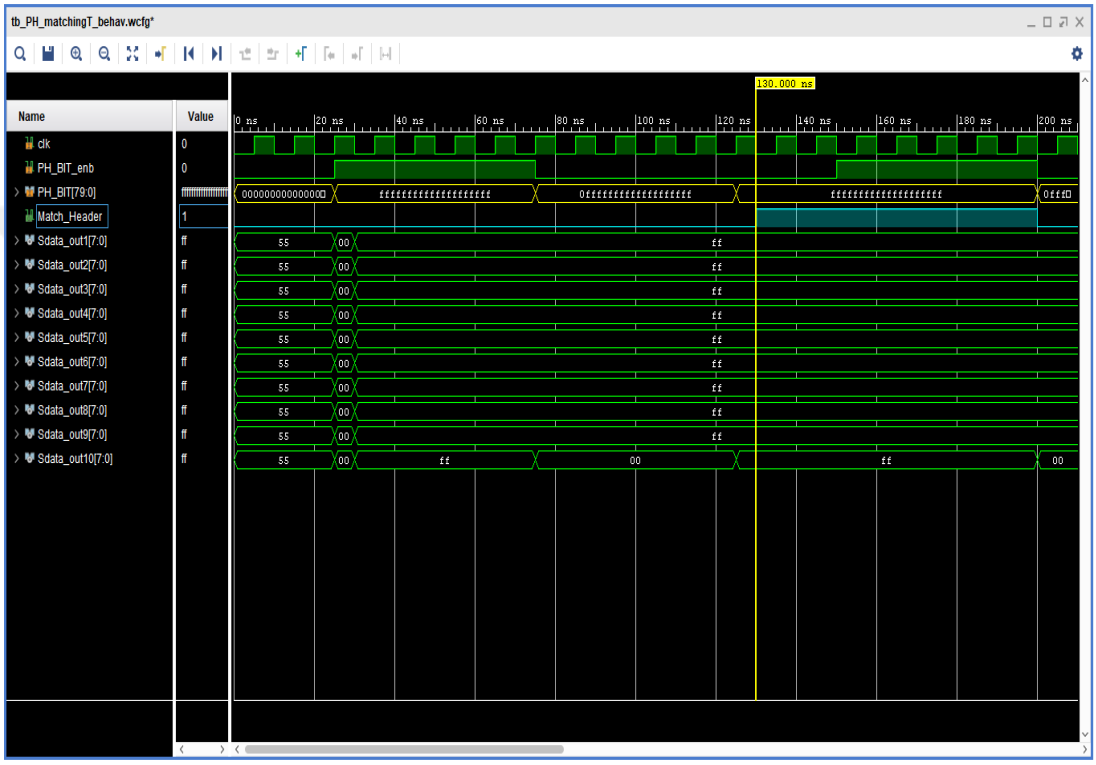

Figure 4.6 The blue color represents the matching simulation result of (PH_BIT) for match result with all 10 RAMs

After the success of the design and implementation of the packet header matching which was used in a fast and efficient way to detect intruders, it prevented them from accessing the data. The two Vivado reports and the project summary can be viewed. The utilization Design Information contains the slice Logic, a summary of Registers, Memory, Clocking, and other information. Table 4.1 shows the Slice Logic information. The design uses a small number of LUT and memories.

Table 4-1 The slice logic information

Name	Slice LUTs (63400)	Slice Registers (126800)	F7 Muxes (31700)	F8 Muxes (15850)	Bonded IOB (210)	BUFGCTRL (32)
N PH_matchingT	355	1	160	80	83	2
comparator (H_Comp...	0	1	0	0	0	0
RAM9 (RAM9T)	36	0	16	8	0	0
RAM8 (RAM8T)	35	0	16	8	0	0
RAM7 (RAM7T)	36	0	16	8	0	0
RAM6 (RAM6T)	35	0	16	8	0	0
RAM5 (RAM5T)	36	0	16	8	0	0
RAM4 (RAM4T)	35	0	16	8	0	0
RAM3 (RAM3T)	36	0	16	8	0	0
RAM2 (RAM2T)	35	0	16	8	0	0
RAM10 (RAM10T)	36	0	16	8	0	0
RAM1 (RAM1T)	35	0	16	8	0	0

The parts of report power analysis and report timing summary show the design used low power and the is no timing delay at the inputs or outputs also, there is zero checking multiple clocks register /latch pins. as a result, this report proved there are 80 input ports with no input delay specified; When checking the output delay, there is one port with no output delay. as shown in Table 4.2,4.3 and Figure 4.7 of the power summary. As expected, all CAM RAMs designed to work in parallel. The time that we need to check all content of RAMs is equal to 5 ns. That means 200 million packet header can process in one second This time is very important in a real-time application or take the right action in time of guide media. Figure 4.8 shown the time of matching processing.

Table 4-2 Part of the power analysis report.

On-Chip	Power (W)	Used	Available	Utilization (%)
Slice Logic	0.813	600	---	---
LUT as Distributed RAM	0.505	320	19000	1.68
LUT as Logic	0.193	35	63400	0.06
F7/F8 Muxes	0.108	240	63400	0.38
BUFG	0.006	2	32	6.25
Others	0.000	2	---	---
Register	0.000	1	126800	<0.01
Signals	1.697	200	---	---
I/O	0.324	83	210	39.52
Static Power	0.103			
Total	2.937			

Table 4-3 Part of the timing delay report.

```
82
83
84  6. checking no_output_delay
85  --------------------------
86   There is 1 port with no output delay specified. (HIGH)
87
88   There are 0 ports with no output delay but user has a false path constraint
89
90   There are 0 ports with no output delay but with a timing clock defined on it or propagating through it
91
92
93  7. checking multiple_clock
94  --------------------------
95   There are 0 register/latch pins with multiple clocks.
96
```

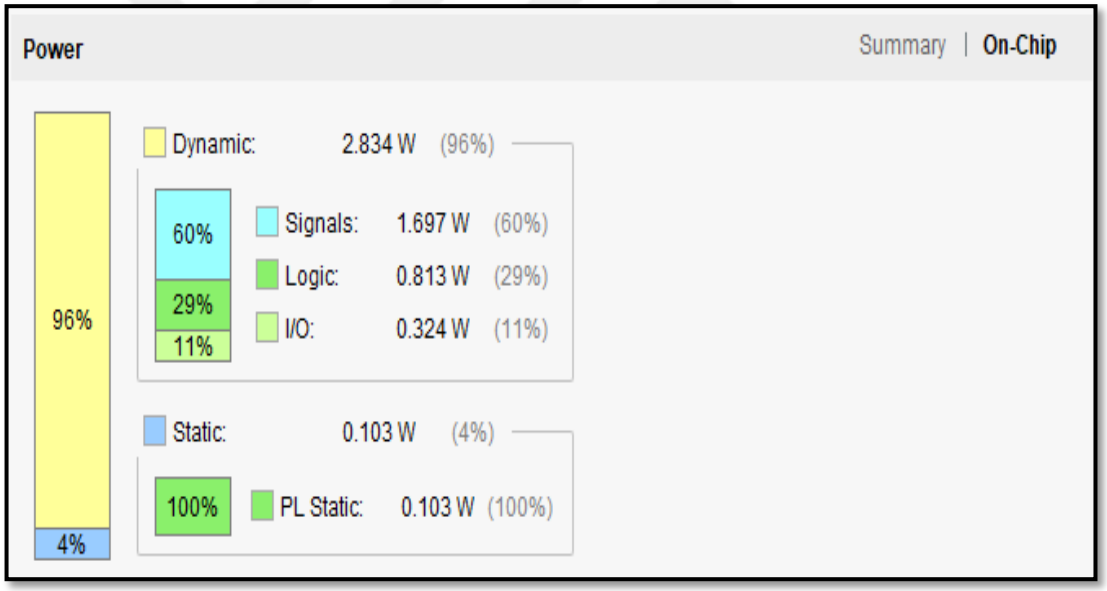

Figure 4.7 Power summary

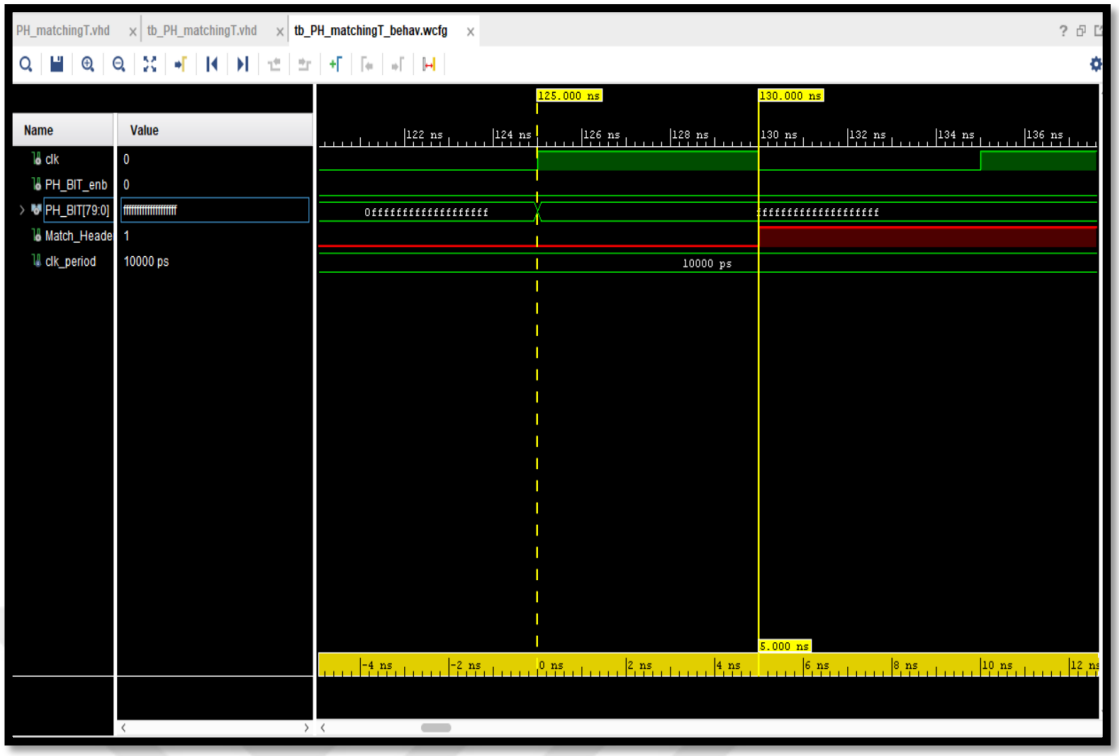

Figure 4.8 The time of matching processing

5 CHAPTER 5

5.1 CONCLUSION AND FUTURE WORK

In this study, was built TCAM based on RAM. We designed hardware architecture of the parallel process multiple RAM that emulates the behaviors of content addressable memory (TCAM) for packet matching to increases speed of detection of intruders on networks and we achieved low power & no Delay in inputs and output with no circuit complexity design. And the small size of the FPGA RAM which is $8*10*2^8$ = 20,480 or 20k bits. use to store every IP header in the world. and this design can process 200 million packet headers in one second

5.2 SUGGESTIONS FOR FUTURE WORK

The following suggestions can be taken into consideration and relied on

as a basis for further future work

1- Use some Rules and these rules can be updated so the mechanism in software-based is also needed to make a more reliable system and to handle more complex rules to detect the intruders on networks
2- In this thesis, we emulated TCAM behaviors that deal with Ipv4 as input for future work that should be focused on the IPv6 networks.
3- decreasing the number of TCAM entries checked in parallel and thus reducing the power consumption. And decreasing the size of memory and the time required to find matching input header packets.
4- for future work can modify the proposed TCAM emulation to be general and can be applied to many applications

6 REFERENCE

[1] T. N. Thinh, T. T. Hieu, and S. Kittitornkun, "A FPGA-based deep packet inspection engine for Network Intrusion Detection System," *Conf. in International on Electrical Engineering/Electronics, Computer, Telecommunications, and Information Technology*, 2012, pp. 1–4.

[2] L. H. Abood, "FPGA Implementation of Ternary Content Addressable Memory," *AL-Mansour Journal,* vol. 24, pp.51-71, 2015.

[3] I. Ullah, Z. Ullah, and J.-A. Lee, "Efficient TCAM design based on multipumping-enabled multiported SRAM on FPGA," *IEEE Access*, vol. 6, pp. 19940–19947, 2018.

[4] Y.-K. Chang and H.-C. Chen, "Fast packet classification using recursive endpoint-cutting and bucket compression on FPGA," *Comput Journal*, vol. 62, pp. 198–214, 2019.

[5] M. Montagna, "Packet Classification Using Growing Hierarchical Self-Organized Map." MSc. thesis University of Illinois at Chicago, 2019.

[6] W. Yu, S. Sivakumar, and D. Pao, "Pseudo-TCAM: SRAM-based architecture for packet classification in one memory access," *IEEE Netw. Lett.*, vol. 1, pp. 89–92, 2019.

[7] Z. Ullah, M. K. Jaiswal, and R. C. C. Cheung, "Z-TCAM: an SRAM-based architecture for TCAM," *IEEE Trans. very large scale Integr. Syst.*, vol. 23, pp. 402–406, 2014.

[8] H. Song and J. W. Lockwood, "Efficient packet classification for network intrusion detection using FPGA," *Conf.Proceedings of the ACM/SIGDA 13th international symposium on Field-programmable gate arrays, 2005*, pp. 238–245.

[9] A. Wicaksana and A. Sasongko, “Fast and reconfigurable packet classification engine in FPGA-based firewall,” *Conf. in Proceedings of the International on Electrical Engineering and Informatics, ICEEI 2011*, pp. 1–6

[10] S. Ezzati, H. R. Naji, A. Chegini, and P. Habibimehr, “Intelligent firewall on reconfigurable hardware,” *Eur Journal. Sci. Res.*, vol. 47, pp. 509–516, 2010.

[11] W. Jiang and V. K. Prasanna, “Large-scale wire-speed packet classification on FPGAs,” *Conf. in Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays*, 2009, pp. 219–228.

[12] W. Jiang and V. K. Prasanna, “Sequence-preserving parallel IP lookup using multiple SRAM-based pipelines,” *Journal .Parallel Distrib. Comput*, vol. 69, pp. 778–789, 2009.

[13] A. Ahmed, K. Park, and S. Baeg, “Resource-efficient SRAM-based ternary content addressable memory,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 25, pp. 1583–1587, 2016.

[14] W. Jiang, “Scalable ternary content addressable memory implementation using FPGAs,” *Conf. in Architectures for Networking and Communications Systems, 2013,* pp. 71–82.

[15] S. M. Qasim, S. A. Abbasi, and B. Almashary, “An overview of advanced FPGA architectures for optimized hardware realization of computation intensive algorithms,” *Conf. in International Multimedia, Signal Processing and Communication Technologies, 2009,* pp. 300–303.

[16] Z. Qian and M. Margala, “Low power RAM-based hierarchical CAM on FPGA,” *in International Conf. on ReConFigurable Computing and FPGAs (ReConFig14), 2014,* pp. 1–4.

[17] K. Garg, A. Moudgil, B. Das, M. F. L. Abdullah, B. Pandey, and D. M. Akbar Hussain, “GTL based internet of things enable processor specific RAM design on 65 nm FPGA,” *Conf in IEEE 57th International Symposium ELMAR. 2015,* pp. 28–30.

[18] Z. Ullah, M. K. Jaiswal, and R. C. C. Cheung, “E-TCAM: An efficient SRAM-based architecture for TCAM,” *Circuits, Syst. Signal Process.*, vol. 33, pp. 3123–3144, 2014.

[19] Chae, Soo-Ik, et al., “Content-Addressable Memory for VLSI Pattern Inspection", *IEEE Journal Of Solid-State Circuits*, vol. 23, pp. 74–78, Feb. 1988.

[20] S. Laurence, “Implementation of ZTCAM using FPGA,” *International Conf. on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT), 2017,* pp. 137–140.

[21] M.-S. Ou, "Implementation and research on improving the efficiency of SDN switch TCAM with ROHC packet compression." MSc thesis National Central University, 2019.

[22] I. Ullah, Z. Ullah, and J.-A. Lee, “Ee-tcam: An energy-efficient sram-based tcam on fpga,” *Electronics Journal* , vol. 7, pp. 2-15, 2018.

[23] Z. Ullah, M. K. Jaiswal, R. C. C. Cheung, and H. K. H. So, “UE-TCAM: an ultra efficient SRAM-based TCAM,” *Conf. in TENCON 2015- IEEE Region 10 2015,* pp. 1–6.

[24] D.-Y. Lee, C.-C. Wang, and A.-Y. Wu, “Bundle-updatable SRAM-based TCAM design for openflow-compliant packet processor,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 27, pp. 1450–1454, 2019.

[25] Z. Ullah, K. Ilgon, and S. Baeg, “Hybrid partitioned SRAM-based ternary content addressable memory,” *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 59, pp. 2969–2979, 2012.

[26] F. Abel, J. Weerasinghe, C. Hagleitner, B. Weiss, and S. Paredes, “An FPGA platform for hyperscalers,” *Conf.IEEE 25th Annual Symposium on High-Performance Interconnects (HOTI) 2017,* pp. 29–32.

[27] Q. Zhao, K. Yanagida, M. Amagasaki, M. Iida, M. Kuga, and T. Sueyoshi, "A logic cell architecture exploiting the shannon expansion for the reduction of configuration memory," *International Conf. on Field Programmable Logic and Applications (FPL), 2014,* pp. 1–6.

[28] Sanjay Churiwala , "Performance evaluation of packet classification on FPGA-based TCAM emulation architectures" IEEE Global Communications *Conf* .(GLOBECOM),2017, pp. 108-98.

[29] Smith, Michael John Sebastian. *Application-specific integrated circuits.* India: University in Kochi,1997, pp. 123-35.

[30] I. Kuon, R. Tessier, and J. Rose, *FPGA architecture: Survey and challenges, foundations and trends r in electronic design automation*. Boston : now Publishers Inc, 2007 pp. 70-3.

[31] P. K. Mishra, T. D. Dongale, R. S. Kamath, and R. K. Kamat, "Soft IP Core based PKI: A Micrologic based Implementation Approach." *International Journal of - Science and Research (IJSRE),* Vol. 4 , pp.5017 -5020 March 2016

[32] M. Arch, "COMPUTER APPLICATIONS" MSc. thesis, Savitribai Phule Pune University, India, 2015

[33] U. Farooq, Z. Marrakchi, and H. Mehrez, *Tree-based heterogeneous FPGA architectures: application specific exploration and optimization*. Springer Science & Business Media, 2012.pp.100-70

[34] R. Munden, *ASIC and FPGA Verification*. London, UK: Springer, 2004, pp. 336-123.

[35] Wah, Ow Shen. "Study on FPGA Based IIR Filter Using Quantitative Approach" MSc. thesis, Universiti Teknologi , Malaysia, 2012

[36] P. Bagdalkar and L. Ali, “Interfacing of light sensor with FPGA using I2C bus,” *International Conf. on Advanced Computing and Communication Systems (ICACCS), 2020*, pp. 843–846.

[37] T. Virtex and X. T. Fpgas, “Virtex-7 T and XT FPGAs Data Sheet : DC and AC Switching Characteristics,” vol. 183, pp. 1–68, 2013.

[38] Xilinx, Vivado Design Suite User Guide , Aug 25,2020 [May,5,2020]

[39] V. Pangracious, Z. Marrakchi, and H. Mehrez, “Two Dimensional FPGAs: Configuration and CAD Flow,” in *Three-Dimensional Design Methodologies for Tree-based FPGA Architecture*, Springer, 2015, pp. 73–94.

[40] S. Brown and J. Rose, “Architecture of FPGAs and CPLDs: A tutorial,” *IEEE Des. Test Comput.*, vol. 13, no. 2, pp. 42–57, 1996.